Probabilistische Fahrzeugumfeldschätzung für Fahrerassistenzsysteme

Von der Fakultät für Elektrotechnik, Informationstechnik, Physik
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation von

Simon Steinmeyer
aus Braunschweig

eingereicht am:	06.01.2014
Disputation am:	13.05.2014
Referenten:	Prof. Dr.-Ing. Markus Maurer
	Prof. Dr.-Ing. Klaus Dietmayer

Druckjahr: 2014

Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen AG.

Kurzbeschreibung

Viele aktuelle Fahrerassistenzsysteme wie beispielsweise die adaptive Geschwindigkeitsregelung, Spurwechselassistenten und Systeme zur Anhaltewegverkürzung sind auf eine verlässliche Detektion anderer Verkehrsteilnehmer und Hindernisse angewiesen. Zukünftige Assistenzsysteme wie beispielsweise Systeme für das Automatische Fahren erhöhen diese Zuverlässigkeitsanforderung weiter.

Die Dissertation befasst sich mit der statistisch genauen Bewertung von Objekthypothesen innerhalb einer Sensordatenfusion, welche aus Messdaten gewonnen wurden. Für jede Hypothese wird eine Wahrscheinlichkeit bestimmt, welche angibt, ob diese vom Fahrerassistenzsystem berücksichtigt werden muss. Hierbei werden widersprüchliche Messdaten systematisch in probabilistischen Modellen aufgelöst, wobei zur Approximation der Wahrscheinlichkeitsdichtefunktion geeignete Modelle aus dem Bereich des Maschinellen Lernens eingesetzt werden. Als Ergebnis erhält man einen Schätzer, der eine präzise Relevanzwahrscheinlichkeit für beliebige Objekthypothesen erzeugt, sodass das Fahrerassistenzsystem frühzeitig und angemessen auf ein aktuelles Umfeld reagieren kann.

Neben dem Objekthypothesenmodell ist als zweiter Typ von Umfeldmodellen das Belegungsgitter verbreitet, welches den Raum um das Fahrzeug in Zellen diskretisiert. Die Messdaten werden mit den jeweiligen örtlich zugehörigen Zellen assoziiert und deren Zustand wird aktualisiert. Als Ergebnis erhält man eine Menge von Zellen mit unterschiedlichen Zuständen, die beispielsweise die Überfahrbarkeit repräsentieren. Die Dissertation entwickelt formale Eigenschaften, die Fusions- und Abfragealgorithmen aufweisen müssen, um eine statistisch belastbare Aussage über die Befahrbarkeit eines aus vielen Zellen bestehenden Korridors liefern zu können. Zusätzlich werden exemplarische Algorithmen entwickelt, die diese Eigenschaften berücksichtigen und somit eine präzisere Schätzung als bekannte Ansätze erlauben.

Abstract

Many of today's driver assistance systems, like adaptive cruise control, lane change assistant or collision avoidance and mitigation systems require a reliable perception of other traffic participants and obstacles. Future driver assistance systems like automatic driving will further increase the requirement of a reliable environment perception.

This thesis deals with the validation of object hypotheses that are generated on the base of measurements inside a sensor data fusion software. A statistically accurate probability of each object hypothesis is generated, which indicates if it should be considered by the driver assistance system. Contradictory data will be resolved systematically using probabilistic models. To approximate the underlying probabilistic density function, proper Machine Learning algorithms are used. As a result, an estimator can be presented that generates a accurate relevance probability for every object hypothesis. Driver assistance systems can now react more early and more adequately to the current environment.

Beside the object model, a second type of environment model is common: The occupancy grid discretises the space around the vehicle into cells, in which each of them contains a cell state. These cell states are updated with measurements that can be associated with the cell's position. As a result, a set of cell states is generated that may represent, for instance, their trafficability. To provide a trafficability estimation of a corridor consisting of many cells, formal mathematical standards are developed. These standards must be considered from both fusion and query algorithms to perform a statistically correct estimation. Additionally, exemplary algorithms with these features are developed which can do a more accurate estimation than common approaches.

Inhaltsverzeichnis

1. Einleitung **1**

1.1. Einleitung und Motivation . . . 1
1.2. Aktuelle Fahrerassistenzsysteme . . . 3
1.3. Zielsetzung dieser Arbeit . . . 5
1.4. Gliederung . . . 6

2. Sensorik **9**

2.1. Sensorik zur Fahrdynamikerfassung des Ego-Fahrzeuges . . . 9
2.1.1. Raddrehzahlsensor . . . 9
2.1.2. Beschleunigungssensor . . . 12
2.1.3. Gierratensensor . . . 13
2.1.4. Inertialplattformen . . . 14
2.1.5. Bewertung . . . 14
2.2. Sensortechnologien zur Umfelderfassung . . . 15
2.2.1. Monokamerasensor . . . 15
2.2.2. Stereokamerasensor . . . 20
2.2.3. Radarsensor . . . 22
2.2.4. Lidarsensor . . . 27
2.2.5. Ultraschallsensoren . . . 29
2.2.6. Bewertung . . . 30
2.3. Zusammenfassung . . . 33

3. Sensordatenfusion **35**

3.1. Einleitung . . . 35
3.2. Anforderungen . . . 36
3.3. Dynamische Zustandsschätzer . . . 38
3.4. Stand der Technik: Sensordatenfusionsarchitekturen . . . 38
3.5. Entwicklung einer Sensordatenfusionsarchitektur . . . 42
3.5.1. Pipe & Filters-Architektur . . . 42
3.5.2. Verwendetes Schichtenmodell . . . 43
3.5.3. Eingabeschicht: Abstraktion der Empfangswege . . . 46
3.5.4. Dekodierungsschicht: Abstraktion durch generische Formate . . . 46
3.5.5. Fusionschicht: Teilfusionen und Dienste . . . 51

3.5.6. Enkodierungsschicht: Trennung von Umfeldmodell und Ausgabeprotokoll 58
3.5.7. Ausgabeschicht: Abstraktion des Sendeweges 60
3.5.8. Ablaufsteuerung 60
3.6. Anwendungen und erreichte Modularität 63
3.7. Zusammenfassung 64

4. Existenz- und Signifikanzschätzung für Umfeldobjekte 67
4.1. Einleitung 67
4.2. Gliederung 68
4.3. Existente und signifikante Objekte 69
4.4. Stand der Technik: Existenzwahrscheinlichkeitsschätzer . . . 70
4.4.1. Track Score Funktion nach Sittler 71
4.4.2. Trackqualität nach Sinha 73
4.4.3. Objektplausibilisierung nach Schoeberl 76
4.4.4. Familie der „Probabilistischen Datenassoziationsalgorithmen“ 78
4.4.5. Entwicklung eines Existenzwahrscheinlichkeitsschätzer mithilfe von zyklischen Markov-Ketten 81
4.4.6. Statische Regeln 85
4.4.7. Bewertung 87
4.5. Stand der Technik: Signifikanzschätzer 89
4.5.1. Bestimmung des Kursverlaufes 90
4.5.2. Bestimmung der Fahrschlauchbreite 91
4.5.3. Probabilistische Fahrschlauchgenerierung und -zuordnung 92
4.5.4. Bewertung 93
4.6. Entwicklung eines integrierten Relevanzschätzer 94
4.6.1. Definition des Zustandsraumes 95
4.6.2. Prospektive Erzeugung der Trainingsdaten 98
4.6.3. Anforderungen und Auswahl von Maschinellen Lernalgorithmen 100
4.6.4. Anwendung der Lernverfahren zur Relevanzklassikation 104
4.6.5. Dimensionsreduktion des Zustandsraumes 105
4.6.6. Tiefpassfilterung des Ergebnisses 109
4.6.7. Auswertung mittels Grenzwertoptimierungskurven . . 110
4.6.8. Histogrammanalyse und -korrektur 112
4.6.9. Bewertung 115
4.7. Ableitung von Regeln aus einem integrierten Relevanzschätzer 117

4.8. Zusammenfassung 123

5. Probabilistische Belegungsgitterfusion 126

5.1. Einleitung 126
5.2. Anforderungen an eine Belegungsgitterfusion 128
5.3. Stand der Technik: Algorithmen und Datenstrukturen für Belegungsgitter 129
5.3.1. Zweidimensionaler Ringpuffer 130
5.3.2. Punktalgorithmus 131
5.3.3. Linienalgorithmus 132
5.3.4. Entwicklung eines Polygonalgorithmus 134
5.4. Grundlagen: Logikalgorithmen für Belegungsgitter 140
5.4.1. Wahrnehmungsrahmen und Massefunktionen 141
5.4.2. Bayes-Logik 142
5.4.3. Dempster-Shafer Theorie 144
5.4.4. Dezert-Smarandache Theorie 146
5.4.5. Zusammenfassung 148
5.5. Stand der Technik: Fusionsalgorithmen für Belegungsgitter . 149
5.5.1. Belegungsgitterfusion nach Elfes 150
5.5.2. Belegungsgitterfusion nach dem MURIEL-Ansatz . . . 155
5.5.3. Belegungsgitterfusion mittels erlernter inverser Sensormodelle 158
5.5.4. Bewertung 164
5.6. Abhängigkeiten in Belegungsgittern 165
5.6.1. Inter-Cell Dependency Problem 165
5.6.2. Inter-Measurement Dependency Problem 169
5.6.3. Kombination von ICDP und IMDP 170
5.6.4. Zusammenfassung 172
5.7. Entwicklung von ICDP-Algorithmen 172
5.7.1. Nichtprobabilistischer Algorithmus 173
5.7.2. Probabilistischer Fusionsalgorithmus 176
5.7.3. Belegungsgitter mit variabler Auflösung 189
5.7.4. Zusammenfassung 192
5.8. IMDP-Algorithmen 195
5.8.1. Übersicht existierender IMDP-Algorithmen 195
5.8.2. Entwicklung eines IMDP-Algorithmus 197
5.8.3. Zusammenfassung 203
5.9. Vergleiche und Anwendungen 204
5.9.1. Vergleich mit klassischem Dempster-Shafer-Ansatz . 204
5.9.2. Echtzeitfähigkeit 206
5.10. Zusammenfassung 208

6. Zusammenfassung und Ausblick **210**
6.1. Zusammenfassung . . . 210
6.2. Ausblick . . . 212

A. Anhang **215**
A.1. Begriffsdefinitionen . . . 215
A.1.1. Echtzeitfähigkeit . . . 215
A.1.2. Statistisch korrekte Schätzer . . . 215
A.2. Kalmanfilter . . . 217
A.3. Maschinelle Lernverfahren . . . 219
A.3.1. Bayessches Netz . . . 219
A.3.2. k-Nearest-Neighbour . . . 220
A.3.3. Lineare Support Vector Machine . . . 221
A.3.4. Support Vector Machine mit Kernelerweiterung . . . 224
A.3.5. Künstliche neuronale Netze . . . 226

Abkürzungsverzeichnis

ABS	Antiblockiersystem
ACC	Geschwindigkeitsregeltempomat (engl. „Adaptive Cruise Control“)
ANN	Künstliches Neuronales Netz (engl. „Artificial Neuronal Network“)
CAN	Ein häufig im Automobilbereich eingesetzter Feldbus (engl. „Controller Area Network“)
CCD	Lichtempfindliche elektronische Bauelemente (engl. „Charge-coupled Device“)
DARPA	Behörde des US-Verteidigungsministerium für Forschungsprojekte (engl. „Defence Advanced Research Projects Agency“)
DSmT	Dezert-Smarandache Logik (engl. „Dezert-Smarandache Theory“)
DST	Dempster-Shafer Logik (engl. „Dempster-Shafer Theory“)
ESC	Fahrdynamikregelung (engl. „Electronic Stability Control“)
FAS	Fahrerassistenzsystem
FMCW	Dauerstrich-Frequenzmodulation-Verfahren (engl. „Frequency Modulated Continuous Wave“)
GPS	Globales Satellitennavigationssystem (engl. „Global Positioning System“)
ICDP	Abhängigkeitsproblem zwischen Zellen (engl. „Inter-Cell-Dependency-Problem“)
IMDP	Abhängigkeitsproblem zwischen Messungen (engl. „Inter-Measurement-Dependency-Problem“)
IMU	Inertialplattform (engl. „Inertial Measurement Unit“)
IPDA	Integrierte Probabilistische Datenassoziation (engl. „Integrated Probabilistic Data Association“)
JIPDA	Vereinheitlichte Integrierte Probabilistische Datenassoziation (engl. „Joint Integrated Probabilistic Data Association“)

JPDA	Vereinheitlichte Probabilistische Datenassoziation (engl. „Joint Probabilistic Data Association“)
kNN	Klassifikationsalgorithmus „k-Nächste-Nachbarn“ (engl. „k-Nearest-Neighbour“)
LLR	Logarithmischer Wahrscheinlichkeitsquotient (engl. „Logarithmic Likelihood Ratio“)
LR	Wahrscheinlichkeitsquotient (engl. „Likelihood Ratio“)
MURIEL	Belegungsgitterfusionsmethode, welche bestimmte Abhängigkeiten berücksichtigt (engl. „Multiple Representation, Independent Evidence Log“)
PDA	Probabilistische Datenassoziation (engl. „Probabilistic Data Association“)
PReVENT	EU-Förderprojekt zur Steigerung der Verkehrssicherheit (engl. „PReVENTive and Active Safety Applications Integrated Project“)
ROC-Kurve	Grenzwertoptimierungskurve (engl. „Receiver Operating Characteristic“)
SDF	Sensordatenfusion
SVM	Stützvektormaschine (engl. „Support-Vector-Machine“)
TTC	Zeit bis zur Kollision (engl. „Time-To-Collision“)
UDP	Ein verbindungsloses Internetprotokoll (engl. „User Datagram Protocol“)
WDF	Wahrscheinlichkeitsdichtefunktion

Kapitel 1.

Einleitung

1.1. Einleitung und Motivation

Das Automobil ist eines der weitverbreitetsten Produkte zur Wahrnehmung individueller Mobilität. Es bietet dem einzelnen Menschen die Freiheit, räumlich jederzeit beweglich zu sein und erhöht dessen Wirkungskreis wesentlich. Der Mensch kann somit aus einer Vielzahl von Arbeits- und Freizeitmöglichkeiten wählen, sodass das Automobil zu seiner persönlichen Entfaltung mit beiträgt.

Zur Nutzung dieser Freiheit muss sich der Fahrer im Gegenzug jedoch der Fahrzeugführungsaufgabe widmen. Diese kann er einerseits als lästige Pflicht empfinden, beispielsweise bei langen Stau- oder Autobahnfahrten. Andererseits beherrscht der Mensch die Regelungsaufgabe nicht perfekt: Durch Fahrfehler und Fehleinschätzungen kombiniert mit dem hohen Schadenspotenzial eines sich schnell bewegenden Kraftfahrzeuges kommt es immer wieder zu Unfällen zum Teil mit erheblichen Folgen für die Beteiligten.

Um die Fahrzeugführung komfortabler und sicherer zu gestalten, entwickeln Fahrzeughersteller verschiedenste technische Systeme: Hierbei befinden sich im aktuellen Fokus der Forschung und Entwicklung insbesondere sog. Fahrerassistenzsysteme (FAS), welche direkt Einfluss auf Fahrzeugregelung oder Fahrer nehmen. Diese haben das Potenzial, den Fahrer aktiv bei anspruchsvollen Fahrsituationen oder eintönigen Aufgaben zu unterstützen, und so einer Überforderung oder einer Unachtsamkeit direkt entgegenzuwirken.

Eine wachsende Anzahl dieser Systeme ist bereits in vielen Fahrzeugen verfügbar: So informieren Navigationssysteme über die kürzeste Route und auftretenden Verkehrsstörungen, Spurhalte- und Fahrstreifenwechselassistent wirken potenziell gefährlichen querdynamischen Fahrmanövern entgegen, der Abstandsregeltempomat (ACC) und Frontalkollisionsschutzsysteme unterstützen den Fahrer bei der Fahrzeuglängsregelung, während das Antiblockiersystem (ABS) und die Fahrdynamikregelung (ESC) in fahr-

dynamischen Grenzsituationen den Fahrer bei der Kontrolle über das Fahrzeug unterstützt. Verschiedenste Studien weisen auch ein hohes Wirkpotenzial für solche Systeme auf das Unfallgeschehen aus [Breuer 2009; Thomas 2007; Bartels 2009].

Neben Aspekten der Sicherheit adressieren Fahrerassistenzsysteme auch Komfortfunktionen: So können Teile der Fahrzeugregelung in bestimmten Situationen vom Fahrer an solch ein System abgegeben werden, sodass der Fahrer diese nur noch überwachen muss. Eine Ausbaustufe ist dabei das „Teilautomatische Fahren“, in der die komplette Fahrzeugführung in bestimmten Domänen und Situationen vom Fahrzeug übernommen wird und der Fahrer nur als Überwachungs- und Rückfallebene zur Verfügung stehen muss [Weiser 2009; Gasser 2012]. Langfristig könnte die Überwachungsaufgabe sogar entfallen, sodass der Fahrer sich mit Nebentätigkeiten während der Fahrt beschäftigen kann. Diese Fahrfunktion wird als „Hochautomatisches Fahren“ bezeichnet, sofern der Fahrer vor dem Erreichen von Systemgrenzen die Fahrzeugführung kurzfristig wieder übernehmen muss. Hat die Fahrfunktion hingegen ausreichend Zeit, bei einer nicht erfolgten Übernahme einen risikominimalen Zustand zu erreichen, bevor Systemgrenzen erreicht werden, so unterstützt die Fahrfunktion „Vollautomatisches Fahren“.

Neben Sicherheit und Komfort können zudem Aspekte der Energieeffizienz bei der Umsetzung einer automatisierten Längsregelung mitintegriert werden. Hierzu kann beispielsweise ein auf den Antriebsstrang optimiertes und vorausschauendes Geschwindigkeitsprofil erstellt werden, welches Straßentopologie, Geschwindigkeitsbegrenzungen und andere Verkehrsteilnehmer berücksichtigt. Zusammen mit Technologien zur intelligenten Verkehrssteuerung können solche Systeme Emissionen verringern und somit einen Beitrag zum Umwelt- und Klimaschutz leisten.

Durch die vielen Möglichkeiten auf das Verkehrsgeschehen positiv einzuwirken, ist bereits ein Trend zu erkennen, der einen stetig wachsenden Wert verkaufter Fahrerassistenzsysteme aufzeigt. Dieser Trend wird einerseits durch den Käufer erzeugt, der insbesondere der Fahrzeugsicherheit einen hohen Wert beimisst, andererseits durch gesetzliche Vorgaben bzw. Selbstverpflichtungen der Automobilindustrie gefördert.

So sind seit 2004 alle neuzugelassenen Autos in der EU und in Japan mit ABS ausgerüstet. Seit November 2011 müssen alle Neuzulassungen innerhalb der EU zusätzlich mit ESC ausgestattet sein; die Vorgabe eines Reifendruckkontrollsystems folgte 2012. Für die meisten in der EU neuzugelassenen schweren Nutzfahrzeuge sind seit 2013 ein Frontalkollisions-

schutzsystem und ein Spurverlassenswarner vorgeschrieben [Gail 2010][1]. Weitere gesetzliche Vorgaben, wie beispielsweise im Bereich des Fußgängerschutzes, sind in Vorbereitung.

Neben Fortschritten im Insassenschutz und der Verbesserung der Verkehrsinfrastruktur sind solche Assistenzsysteme im Bereich der Fahrdynamikstabilisierung mit dafür verantwortlich, dass sich die Anzahl der Verkehrsunfälle bzgl. Schwerverletzten und Toten in der EU trotz höheren Verkehrsaufkommens seit Jahren rückläufig entwickelt. Bei der langfristigen Fortsetzung dieses positiven Trends werden Fahrerassistenzsysteme eine wichtige Rolle spielen, welche eine Umfeldwahrnehmung beinhalten und somit eine Re- und Interaktion mit dem Fahrzeugumfeld ermöglichen.

Zurzeit ist die Ausstattungsrate solcher Systeme im bundesdeutschen Flottendurchschnitt jedoch eher gering, sodass noch ein hohes Potenzial aktueller und zukünftiger Systeme zur Verbesserung der Sicherheit, der Energieeffizienz und des Komforts besteht.

1.2. Aktuelle Fahrerassistenzsysteme

Neben den bereits recht weitverbreiteten Systemen zur Fahrdynamikstabilisierung sind in einigen Fahrzeugen auch verschiedenste Fahrerassistenzsysteme verfügbar. Die meisten dieser Systeme wurden in Oberklassefahrzeugen eingeführt, jedoch ist bereits eine Demokratisierung erkennbar, sodass diese mittlerweile auch in anderen Fahrzeugklassen angeboten werden.

Ein wichtiger Bestandteil dieser Systeme ist eine Umfeldwahrnehmung und dessen Interpretation, welche wichtige Eingangsgrößen liefert. Je nach System werden neben einer Eigenbewegungsschätzung zusätzlich Informationen über andere Hindernisse, Verkehrsteilnehmer oder auch über Verläufe bestimmter Fahrstreifen benötigt. Eine Übersicht einiger Fahrerassistenzsysteme und deren benötigten und leistungssteigernden optionalen Umfeldentitäten ist in Tabelle 1.1 dargestellt.

Bei vielen Fahrerassistenzsystemen stellt die Leistungsfähigkeit der Umfeldwahrnehmung und der Interpretation einen limitierenden Faktor dar, welcher bei der Auslegung dieser Systeme in Form von Funktionsgrenzen mitberücksichtigt wird: So beachten viele ACC-Systeme keine oder nicht alle statischen Objekte und limitieren die maximale Bremsverzögerung um Auswirkungen von Fehldetektionen abzuschwächen. Assistenzsysteme zur Vermeidung von Frontalkollisionen arbeiten nur in einem be-

[1] Bei vielen der hier beschriebenen gesetzlichen Regelungen sind Übergangsfristen und Ausnahmegenehmigungen noch zu beachten.

	Einparkassistenz	Abstandsregeltempomat	Frontalkollisionsschutzsystem[2]	Fahrstreifenhalteassistenz	Fahrstreifenwechselassistenz	Fernlichtassistenz
Eigenbewegungs-schätzung	•	•	•	•	•	•
Erkennung stehender Objekte	•	(•)	(•)			•
Erkennung bewegender Objekte	(•)	•	(•)		•	•
Fahrstreifenerkennung		(•)		•	(•)	

Tabelle 1.1.: Benötigte und optionale Umfeldentitäten für bestimmte Fahrerassistenzsysteme

stimmten Geschwindigkeitsbereich und vermeiden heute nicht alle Kollisionen. Viele Spurwechselassistenten überwachen nur den toten Winkel und informieren nicht über schnell überholende Fahrzeuge. Heutige Fahrstreifenhalteassistenten deaktivieren sich, wenn sie auf eine Straßen ohne Fahrstreifenmarkierungen treffen und Fernlichtassistenzsysteme berücksichtigen nicht alle Verkehrsteilnehmern wenn sie keine ausreichende Beleuchtung aufweisen.

Unisono weisen praktisch alle Fahrzeughersteller deshalb darauf hin, dass der Fahrer die alleinige Verantwortung für die Fahrzeugführung trägt, die Fahrerassistenzsysteme nur unterstützend ausgelegt sind und möglicherweise nicht in allen Situationen funktionieren. Aktuelle FAS sind deshalb so ausgelegt, dass der Fahrer sie jederzeit übersteuern und so seiner Verantwortung immer nachkommen kann.

[2] Ein Frontalkollisionsschutzsystem muss je nach Auslegung zumindest stehende oder sich bewegende Objekte erkennen können.

1.3. Zielsetzung dieser Arbeit

Bereits heute haben einige Fahrerassistenzsysteme hohe Anforderungen an die Genauigkeit, Robustheit und Umfang der Umfeldwahrnehmung. Es ist abzusehen, dass zukünftige Fahrerassistenzsysteme diese Anforderungen weiter erhöhen. Dies wird insbesondere bei Systemen der Fall sein, in der der Fahrer keine dauerhafte Überwachungsaufgabe hat, also bei hoch- und vollautomatisierten Fahrfunktionen. Auch kann die Leistungsfähigkeit aktueller Fahrerassistenzsysteme durch eine verbesserte Umfeldwahrnehmung gesteigert werden.

Eine Umfeldwahrnehmung basiert jedoch auf Messdaten, welche von verschiedenen Sensoren erzeugt werden und je nach Messprinzip zu einem gewissen Maß immer ungenau sind. Daneben können Witterungseinflüsse und situationsbezogene Faktoren Messungen negativ beeinflussen, sodass eine ungenaue Zustandserfassung einzelner Umfeldentitäten möglich ist. Dieses kann die Umfeldwahrnehmung insgesamt beeinträchtigen und somit die Leistungsfähigkeit von Fahrerassistenzsystemen verschlechtern: Beispielsweise können andere Fahrzeuge fälschlicherweise dem Fahrstreifen des eigenen Fahrzeuges[3] zugeordnet werden, falls die Fahrbahnschätzung oder Fahrzeugpositionsschätzung ungenau ist.

Des Weiteren können Fehlinterpretationen der Messergebnisse dazu führen, dass wesentliche Entitäten übersehen oder sogar Fehlziele im Sinne des Fahrerassistenzsystems generiert werden. So können unterfahrbare Schilderbrücken als stationäre Hindernisse wahrgenommen werden, wenn sie von einem Radarsensor ungünstig erfasst werden. Ebenso können auch metallische Getränkedosen als relevantes Ziel fehlinterpretiert werden, obwohl diese überfahrbar sind. Umgekehrt können wesentliche Umfeldobjekte mit einem sehr geringen Radarquerschnitt nicht oder erst sehr spät durch Radarsensor wahrgenommen werden.

Um die Umfeldwahrnehmung zu verbessern, ist die Weiterentwicklung von Sensortechnologien möglich, sodass diese idealerweise hochgenau messen und alle Merkmale korrekt interpretieren ohne Fehlziele zu erzeugen und ohne wesentliche Ziele übersehen. Fortschritte in Sensortechnologien werden voraussichtlich Verbesserungen erreichen, jedoch scheint die Entwicklung eines solchen idealen Sensors in naher Zukunft unrealistisch zu sein.

Eine andere Möglichkeit die Umfeldwahrnehmung zu verbessern, besteht in der Fusionierung von Messdaten unterschiedlicher Sensoren und die systematische Auflösung von Messdifferenzen und Widersprüchen. Ein

[3] Das eigene Fahrzeug aus Sicht des Fahrerassistenzsystems wird im folgenden Ego-Fahrzeug genannt

probabilistisches Umfeldmodell kann helfen, diese negativen Einflüsse statistisch genau zu quantifizieren und zu minimieren. Die Entwicklung eines solchen Umfeldmodells ist Ziel dieser Arbeit.

Die in dieser Arbeit entwickelte Umfeldwahrnehmung zielt dabei auf kein bestimmtes Fahrerassistenzsystem ab. Stattdessen versucht die Umfeldwahrnehmung das Verkehrsgeschehen, Fahrbahn und Befahrbarkeit im Rahmen der verfügbaren Daten so genau und aktuell wie möglich zu schätzen. Konkrete Fahrerassistenzapplikationen[4] können dann die benötigten Größen aus dem Umfeldmodell extrahieren und verwenden, sodass die Umfeldwahrnehmung universell verschiedenen Fahrerassistenzsystemen dienen kann.

1.4. Gliederung

Um ein probabilistisches Umfeldmodell zu entwickeln, ist zunächst die gesamte Wahrnehmungskette zu betrachten, wobei frühe Störeinflüsse und zu einfache Modellannahmen identifiziert und ggf. durch bessere Modelle und Schätzer ausgetauscht werden müssen. Diese Wahrnehmungskette wird sukzessiv in den einzeln Kapiteln untersucht, was als Übersicht in Abbildung 1.1 dargestellt ist.

Hierzu werden in Kapitel 2 verschiedene aktuelle Sensortechnologien vorgestellt, wobei sowohl Sensoren zur Erfassung der Eigenbewegung als auch Sensoren zur Umfeldwahrnehmung untersucht werden. Hierbei werden diverse mögliche Störquellen für das jeweilige Messprinzip bzw. bei der Interpretation der Messergebnisse aufgezeigt. Diese können die Basisursache in einer Fehlerkette sein, wenn eine Umfeldschätzung zu ungenaue oder fehlerhafte Ergebnisse liefert.

Im anschließenden Kapitel 3 wird eine exemplarische Sensordatenfusion vorgestellt, welche u. a Objekte, Hindernisse, Befahrbarkeit, Eigenbewegung sowie Fahrbahnverlauf und -eigenschaften schätzt. Hierzu werden die Messdaten von diversen Sensoren in verschiedenste Teilschätzer eingebracht, sodass ein Umfeldmodell mit hohem Detaillierungsgrad erzeugt wird. Aus diesem lassen sich die benötigten Größen für verschiedenste Fahrerassistenzapplikationen extrahieren.

Innerhalb der Sensordatenfusion werden unterschiedliche Modelle verwendet, um Umfeldobjekte zu berücksichtigen: Ein filterbasierter Objekttracker verfolgt und schätzt den Zustand anderer Verkehrsteilnehmer, wo-

[4] Die Fahrerassistenzapplikation ist der Teil eines FAS, der aufgrund eines aufgetretenen Ego- und Umfeldzustandes agiert. Das Erfassen und Fusionieren von Sensordaten gehört nicht dazu.

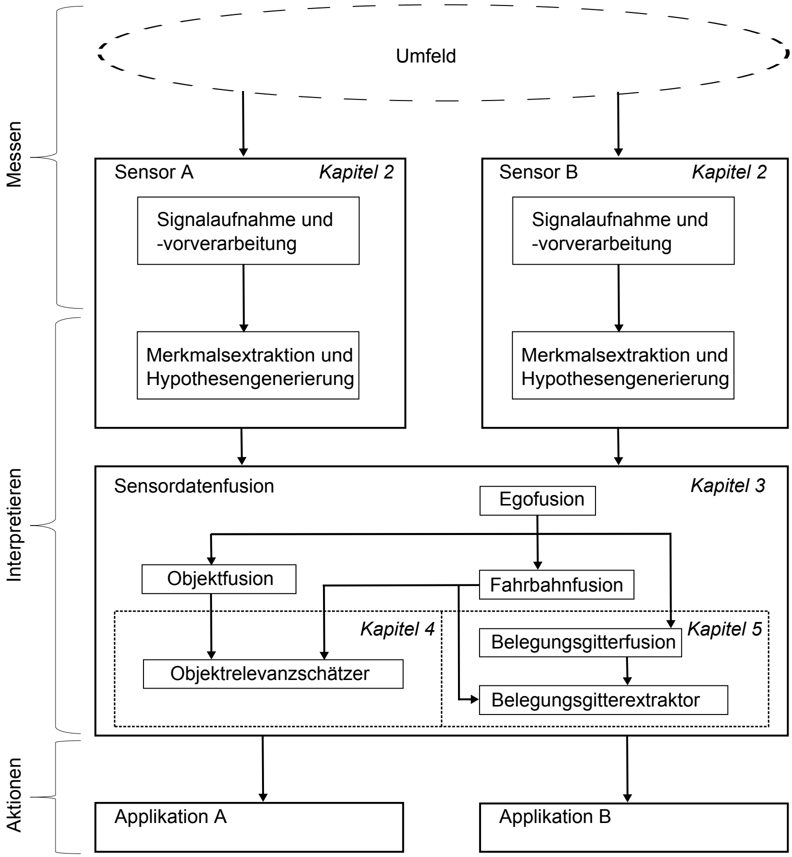


Abbildung 1.1.: Gliederung: Kapitel 4 und 5 (gepunktete Bereiche) sind Schwerpunkte dieser Arbeit

bei viele Randbedingungen modellhaft erfasst werden. Als Ergebnis erhält man eine Liste von Objekthypothesen, welche verschiedene Unsicherheiten beinhaltet: Einerseits können in dieser Liste Scheinobjekte auftreten, andererseits können durch Fehler in der Zustandsschätzung Objekte fälschlicherweise als wichtig bzw. unwichtig im Sinne der Fahrerassistenzapplikation erscheinen.

In Kapitel 4 wird für die Bewertung einer solchen Objektliste ein integrierter Schätzer vorgestellt, der eine kombinierte Existenz- und Signifi-

kanzwahrscheinlichkeit für Objekte ermittelt und dabei sämtliche Abhängigkeiten der Informationsquellen berücksichtigt. Somit werden beispielsweise Unsicherheiten, Ungenauigkeiten und Fehler von Messprinzipien, Signalvorverarbeitungen, Hypothesengenerierung, Interpretation, Modellannahmen, Umwelteinflüssen und sonstigen Quellen in den probabilistischen Schätzer mit einbezogen. Als Ergebnis erhält die Fahrerassistenzapplikation für eine aktuelle Objektliste genaue Relevanzwahrscheinlichkeiten. Beispielhaft wird dies für einen Relevanzklassifikator gezeigt, der hochautomatisches Fahrfunktionen auf Autobahnen und autobahnähnlichen Straßen unterstützt.

Als Alternative zu der stark modellbehafteten Schätzung eines Objekttrackers und des integrierten Relevanzschätzers verwenden viele Sensordatenfusionen ein sog. „Belegungsgitter“, welches eine modellärmere Umfeldrepräsentation darstellt. Das Umfeld wird dabei in Zellen diskretisiert, welche meist einen Zustand zur Überfahrbarkeit kapseln. In Kapitel 5 wird ein neuer Belegungsgitteransatz vorgestellt, der höhere Anforderungen an Genauigkeit und Robustheit als bekannte Ansätze bewältigen kann. Die Besonderheit ist, dass er Abhängigkeiten zwischen allen Zellzuständen und Messungen deutlich genauer berücksichtigt. Im Ergebnis kann eine Fahrerassistenzapplikation einen beliebigen Bereich im Belegungsgitter abfragen und eine genaue Befahrbarkeitsschätzung erhalten.

Zuletzt werden in Kapitel 6 die vorgestellten Ansätze und Ergebnisse dieser Arbeit zusammengefasst und es wird ein Ausblick auf mögliche zukünftige Entwicklungen im Bereich der Fahrerassistenzsysteme gegeben.

Teile dieser Arbeit sind bereits in Vorabveröffentlichungen dokumentiert: So ist eine Basisbeschreibung der verwendeten Sensordatenfusionsarchitektur in [Steinmeyer 2010a] beschrieben, in [Steinmeyer 2010b] sind die entwickelten Algorithmen zur Relevanzschätzung aus Kapitel 4 veröffentlicht und einige Aspekte zur Belegungsgitterfusion aus Kapitel 5 sind in [Steinmeyer 2009] wiederzufinden.

Kapitel 2.

Sensorik

Fahrerassistenzsysteme benötigen sowohl Informationen über den Dynamikzustand des Ego-Fahrzeuges als auch über das Umfeld. Hierzu werden Odometrie- und Umfeldsensoren eingesetzt, welche meist zyklisch entsprechende Messdaten erzeugen. Je nach Mess- und Auswertungsprinzip können hier verschiedene Quellen identifiziert werden, die Ungenauigkeiten und Fehlinterpretationen verursachen können. Diese werden in diesem Kapitel für die einzelnen Sensoren dargestellt.

2.1. Sensorik zur Fahrdynamikerfassung des Ego-Fahrzeuges

Für viele Fahrerassistenzsysteme ist die Bestimmung der Fahrzeugeigenbewegung grundlegende Voraussetzung für deren Funktion. Hierbei sind je nach Fahrerassistenzapplikation die relative Position, die Längs- und Quergeschwindigkeiten bzw. -beschleunigungen wie auch Gierwinkel und Gierrate relevant. In diesem Abschnitt werden einige Sensoren vorgestellt, die eine Schätzung dieser Größen erlauben.

2.1.1. Raddrehzahlsensor

Einer der wichtigsten Sensoren zur Eigenbewegungsschätzung ist der sog. „Raddrehzahlsensor“, welcher in diskreten Schritten misst, inwieweit sich ein Rad weiterbewegt hat. Ein Steuergerät zählt die jeweiligen Impulse der einzelnen Räder und informiert andere Steuergeräte zyklisch über den aktuellen Stand.

Heutige Raddrehzahlsensoren sind meist aktive Sensoren, welche nach dem sog. „Hallprinzip“ messen und welche schematisch in Abbildung 2.1 dargestellt sind: Bei diesen Sensoren ist am zu messenden Rad ein Polring angebracht, welcher in verschiedene gleichgroße Sektoren unterteilt ist. Benachbarte Sektoren weisen eine unterschiedliche Magnetisierung

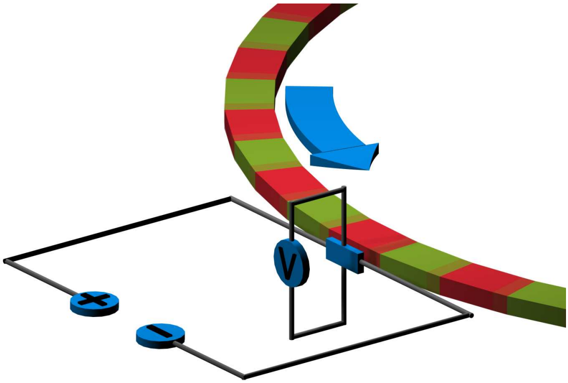


Abbildung 2.1.: Schematischer Aufbau eines Raddrehzahlsensors bestehend aus einem Polring und einem Hall-Element

(Nordpol bzw. Südpol) auf. Das Sensorelement besteht aus einer Leiterplatte, welche mit einer Spannung versorgt wird. Wird diese Leiterplatte einem Magnetfeld ausgesetzt, so kann man dies in Form einer Spannungsänderung quer zur Leiterplatte feststellen. Die unterschiedliche Magnetisierung erzeugt bei einer Radbewegung eine Schwingung, wobei dessen Frequenz proportional zur Geschwindigkeit ist. Für genauere Informationen zu Raddrehzahlsensoren sei auf [Zabler 2010, S. 63 ff] verwiesen.

Unter der Annahme, dass das Ego-Fahrzeug die Strecke zurückgelegt hat, die die gemessenen Radimpulse suggerieren, lassen sich viele verschiedene Größen des Ego-Zustands ableiten. Für die einfache Bestimmung sind insbesondere die Hinterräder geeignet, wenn diese keine aktive Hinterradlenkung aufweisen. Somit sind nur die beiden Zählervariablen i_r bzw. i_l für das rechte bzw. linke Hinterrad die einzigen beiden notwendigen Messwerte zur Eigenbewegungsschätzung.

Als zusätzliche Datenquelle bzw. Apriori-Wissen sind ein Zeitgeber zur Messung der Zeit t und das Wissen über die Anzahl der Impulse pro Radumdrehung N, die effektive Spurweite l und die Radumfänge d_r bzw. d_l des rechten und linken Hinterrades notwendig. Daraus lassen sich die zurückgelegte Strecke s, die Gesamtgeschwindigkeit v, die Beschleunigung a, der Gierwinkel ψ und die Gierrate $\dot{\psi}$ bestimmen.

Der Gierwinkel und die relative Positionsänderungen werden mithilfe eines ortsfesten Koordinatensystems referenziert, dessen Ursprung durch

Größe	wird bestimmt aus	Formel
Zurückgelegte Strecke	i_r, i_l, d_l, d_r, N	$s = \frac{1}{2}\left(\frac{i_r}{N}d_r + \frac{i_l}{N}d_l\right)$
Geschwindigkeit	s, t	$v = \frac{\Delta s}{\Delta t}$
Beschleunigung	v, t	$a = \frac{\Delta v}{\Delta t}$
Gierwinkel	i_r, i_l, d_l, d_r, N, l	$\psi = \frac{1}{l}\left(\frac{i_r}{N}d_r - \frac{i_l}{N}d_l\right)$
Gierrate	ψ, t	$\dot{\psi} = \frac{\Delta\psi}{\Delta t}$
Positionsänderung (x)	s, ψ	$\Delta p_x = \Delta s \cos(\psi)$
Positionsänderung (y)	s, ψ	$\Delta p_y = \Delta s \sin(\psi)$

Tabelle 2.1.: Aus Messungen von Raddrehzahlsensoren ableitbare Größen

Startposition und -ausrichtung des Ego-Fahrzeuges festgelegt wird. Somit betragen die Initialgrößen von Gierwinkel, X- und die Y-Position anfangs Null. Die Bestimmung der einzelnen Größen ist in Tabelle 2.1 dokumentiert.

Die Genauigkeit der geschätzten Größen hängt stark von der fahrdynamischen Situation ab. So ist ein Schlupf der Räder in diesem Modell genauso wenig berücksichtigt, wie ein auftretender Schwimmwinkel. Insgesamt eignet sich das Modell deshalb vor allem für Fahrsituationen, in denen möglichst keine fahrdynamisch anspruchsvolleren Ereignisse auftreten.

Des Weiteren ist eine möglichst genaue Bestimmung der Radumfänge d_r bzw. d_l und der effektiven Spurweite l notwendig, um die Schätzgrößen zu bestimmen. Die effektive Spurweite kann sich jedoch beispielsweise mit dem Luftdruck und dieser mit der Reifentemperatur ändern. Auch ist die effektive Spurweite nicht ohne Weiteres genau bestimmbar, da der Abstand zwischen den Auflagepunkten der Reifen wesentlich ist. Der Auflagepunkt liegt jedoch auf einem unbekannten Ort auf der Radlauffläche. Nimmt man approximativ den Mittelpunkt der Lauffläche als Mittelwert, so kann der Radstand bis zu der Auflageflächenbreite eines Reifens abweichen. Es ist auch denkbar, dass die Positionen der Auflagepunkte während einer Fahrt variieren und dabei nicht unbedingt symmetrisch ausfallen. Eine weitere mögliche Einflussgröße ist die Querbeschleunigung.

2.1.2. Beschleunigungssensor

Die Längs- und Querbeschleunigung lässt sich zwar aus den Raddrehzahlsensoren bestimmen (s. Tabelle 2.1), jedoch unterliegt diese verschiedenen Störgrößen wie beispielsweise dem Radschlupf. Diese treten verstärkt in fahrdynamischen Grenzsituationen auf, bei denen hohe Beschleunigungs- oder Verzögerungsraten erreicht werden.

Sowohl bei stärkeren Bremssituationen als auch bei sich durchdrehenden Reifen repräsentiert die Strecke, die die Reifen gerollt sind, nicht mehr die Strecke, die das Fahrzeug zurückgelegt hat. Somit wird die zurückgelegte Strecke ungenau bestimmt und damit alle von ihr abgeleiteten Größen, wie beispielsweise die Längs- und Quergeschwindigkeit und entsprechende Beschleunigungen. Zur genaueren Bestimmung dieser Größen in solchen Szenarien können zusätzliche Beschleunigungssensoren helfen, welche die Beschleunigung in X- und Y-Richtung des Fahrzeuges messen. Durch die zeitliche Integration der Beschleunigung hat man eine weitere Quelle für die Geschwindigkeitsänderung, welche unempfindlicher in solchen Fahrsituationen ist.

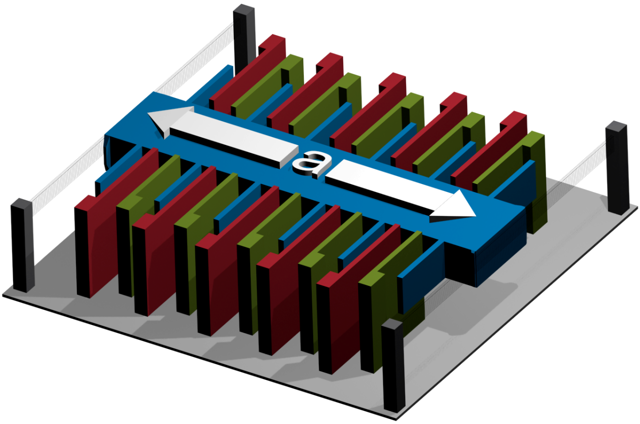


Abbildung 2.2.: Schematisch dargestellter Beschleunigungssensor bestehend aus einer seismischen Masse (blau) und verschiedenen Kapazitäten (rot, grün)

Ein mögliches Messprinzip eines mikromechanischen Beschleunigungssensors ist in Abbildung 2.2 dargestellt: Er basiert auf einer seismischen

Masse, welche sich bei Bewegung anderen Kondensatorplatten nähert bzw. entfernt. Dies bewirkt eine kapazitive Änderung, aus der die Beschleunigung abgeleitet werden kann. Für genauere Informationen zum physikalischen Messprinzip sei auf [Zabler 2010, S. 75 ff] verwiesen.

Für die Bestimmung der Geschwindigkeit bzw. der zurückgelegten Strecke ist es zwar prinzipiell möglich, die Beschleunigung über die Zeit ein- oder zweimal zu integrieren. Hierbei gilt jedoch, dass sich die Messfehler mitintegrieren und Geschwindigkeit und zurückgelegte Strecke somit sehr ungenau werden können.

Als alleiniger Odometriesensor zur Eigenbewegungsschätzung scheidet dieser somit aus, jedoch kann er insbesondere in Situationen, wo starke Beschleunigungen auftreten, die Dynamikschätzung basierend auf Raddrehzahlsensoren verbessern.

2.1.3. Gierratensensor

Die Gierrate kann wie die Beschleunigung in fahrdynamisch anspruchsvolleren Situationen nur unzureichend aus dem Raddrehzahlsensor bestimmt werden. Für Assistenzsysteme im Bereich der Fahrdynamikstabilisierung ist jedoch die Bestimmung dieser Größe in genau solchen Situationen unabdingbar. Hierbei ist insbesondere das elektronische Stabilitätsprogramm (engl. „Electronic Stability Control (ESC)“) zu nennen, welches ein Schleudern des Fahrzeuges erkennen muss.

Als alternativer Sensor eignet sich ein mikromechanischer Drehratensensor, welcher bei geeigneter Positionierung im Fahrzeug dessen Drehrate um die Z-Achse und somit dessen Gierrate misst. Innerhalb eines solchen Drehratensensors befindet sich eine seismische Masse, welche durch einen Aktor in eine oszillierende Schwingung versetzt wird. Bei einer Drehbewegung wird diese seismische Masse einer Corioliskraft ausgesetzt, die die Schwingungsrichtung beeinflusst. Eine Richtungsänderung und dessen Stärke kann durch einen Beschleunigungssensor erfasst werden, der die Beschleunigung senkrecht zur ursprünglichen Schwingungsrichtung misst (vgl. [Zabler 2010, S. 130 ff].

Heutige in Fahrzeugen verbaute Beschleunigungs- und Gierratensensoren werden hauptsächlich als Eingangsgrößen für die Fahrdynamikstabilisierung verwendet, welche nur größere Abweichungen zwischen dem vom Fahrer gewünschten und tatsächlichen Kurs des Ego-Fahrzeuges feststellen muss. Das zugrundeliegende Modell bestimmt dabei Schlupf einzelner Räder, Kurs- und Gierwinkel des Fahrzeuges mit Hilfe von Messungen des Lenkradwinkelsensor und Odometriesensoren (vgl. [von Zanten

2009]). Für Messungen zwecks genauer Eigenbewegungsschätzung sind diese Sensoren jedoch nicht ausgelegt, weswegen ein beachtliches Messrauschen festgestellt werden muss, welches beispielsweise durch Fahrzeugvibrationen und Unebenheiten der Fahrbahn verursacht wird.

Aus der zeitlichen Integration der Gierrate $\dot{\psi}$ kann zwar der Gierwinkel ψ bestimmt werden, jedoch integrieren sich auch hier kleine Schätzfehler. Zu Ungenauigkeiten kommt es insbesondere dann, wenn kein exakter Nullpunktabgleich erfolgt und die Gierrate somit einen systematischen Fehler aufweist. Dies kann unter anderem stärkere Ungenauigkeiten bei der relativen Positionsschätzung hervorrufen.

2.1.4. Inertialplattformen

Inertialplattformen (engl. „Inertial Measurement Unit (IMU)“) integrieren meist mehrere höherwertige Beschleunigungs- und Gierratensensoren für alle drei Raumachsen und können somit eine relative Bewegung ermitteln. Einige IMUs weisen zusätzlich Empfänger auf, welche Signale globaler Satellitennavigationssysteme und ggf. entsprechende Korrekturdaten aus anderen Quellen auswerten können.

Durch diese globale Referenzierung gelingt eine Positionsbestimmung teilweise bis in den Zentimeterbereich, was eine genaue Eigenbewegungsschätzung deutlich vereinfacht. Bei Verwendung mehrerer Antennen ist zudem die direkte Messung der Fahrzeugausrichtung möglich, sodass Gier- und sogar Schwimmwinkel leicht bestimmbar sind. Auch ermöglicht die genaue globale Positionierung eine ständige Kalibrierung der Odometriesensoren, sodass beispielsweise ein Nullpunktabgleich schnell und genau erfolgen kann.

Insgesamt erreichen solche Inertialplattformen zwar eine sehr gute Positions- und Dynamikschätzung des Ego-Fahrzeuges, jedoch eignen sie sich aus Kostengründen momentan nicht für serientaugliche Fahrerassistenzsysteme. Sie sind aber als Referenzsysteme im Forschungs- und Entwicklungsbereich prädestiniert.

2.1.5. Bewertung

In heutigen Serienfahrzeugen sind bereits viele Odometriesensoren verbaut, die eine Eigenbewegungsschätzung erlauben. Sind bestimmte Kalibrationsdaten bekannt, so ist in fahrdynamisch wenig anspruchsvollen Situationen die Auswertung der Radimpulse vielversprechend. Bei starken Beschleunigungs- oder Bremsmanövern repräsentieren diese hinge-

gen nicht mehr die vom Ego-Fahrzeug zurückgelegte Strecke, wobei dann auf mäßig genaue Beschleunigungs- und Gierratensensoren zurückgegriffen werden muss.

Insgesamt ist bei der Fusionierung dieser Odometriedaten mit einem Schätzfehlern zu rechnen, wobei hier insbesondere der Gierwinkel hervorzuheben ist. Einerseits wird dieser nur durch Integration von bestimmten Schätzgrößen ermittelt, wobei sich Fehler mitintegrieren, andererseits können bereits kleine Gierwinkelfehler laterale Scheinbewegungen von Objekten in größerer Entfernung verursachen und die Assoziation von Objekthypothesen erschweren.

Abhilfe schaffen natürlich bessere Sensoren, welche bereits in verschiedenen Kosten- und Genauigkeitsabstufungen innerhalb fertiger Inertialplattformen verfügbar sind und um Größenordnungen genauere Schätzungen erlauben. Um die Verbreitung künftiger Fahrerassistenzsysteme zu fördern, sollte jedoch möglichst auf solche aufwendigen Plattformen verzichtet und versucht werden, mit der Schätzgüte vorhandener Sensoren umzugehen.

2.2. Sensortechnologien zur Umfelderfassung

Für Fahrerassistenzsysteme ist eine Erfassung des Fahrzeugumfeldes notwendig, wobei je nach Fahrerassistenzapplikation verschiedenste Aspekte berücksichtigt werden müssen: Einige benötigen Informationen anderer Verkehrsteilnehmer und Hindernisse, aber auch Fahrstreifenverläufe, Verkehrszeichen und erkannte Gefahrenstellen können von Interesse sein. Eine solche Umfelderfassung ist nur mittels zusätzlicher Sensorik möglich, welche in diesem Abschnitt vorgestellt wird.

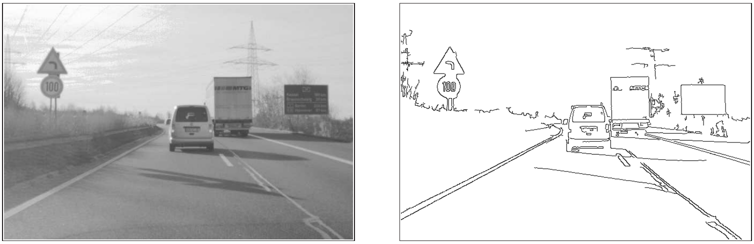
2.2.1. Monokamerasensor

Monokamerasensoren messen mithilfe einer Optik und eines lichtempfindlichen CCD- oder CMOS-Sensors einstrahlendes Licht. Hierbei werden Helligkeiten, Azimut- und Elevationswinkel und je nach Funktion auch Farbwerte erfasst, sodass eine zweidimensionale Projektion eines Umgebungsausschnitts entsteht. Es können jedoch keine Tiefen- oder Geschwindigkeitsinformationen direkt gemessen werden.

Mithilfe bildverarbeitender Algorithmen wird versucht, aus diesen Bildern bestimmte Größen zu extrahieren, welche für ein Fahrerassistenzsystem hilfreich sein könnten. Hierzu gibt es zwei unterschiedliche Methoden:

Die erste Methode ist stark modellbehaftet und sucht Entitäten bekannter Klassen im gegebenen Bild aufgrund bekannter Merkmale. Dazu gehören beispielsweise:

- Fahrstreifenmarkierungen
- Verkehrszeichen
- Andere Verkehrsteilnehmer einer definierten Klasse (z. B. Autos, Fußgänger)



(a) Originalbild mit Fahrstreifenmarkierungen, Teernähten, Verkehrszeichen, Abschattungen und Fahrzeugen

(b) Die Anwendung des Canny-Operators liefert viele mögliche Hypothesen

Abbildung 2.3.: Anwendung des Canny-Kantendetektionsalgorithmus [Canny 1987] auf eine Verkehrsszene (Bundesautobahn 39, nahe Wolfsburg)

Hierzu werden zunächst recht einfache Bildoperatoren angewendet, die bestimmte allgemeine Merkmale extrahieren, welche sich auch in der zu detektierenden Objektklasse wiederfinden. Für eine Fahrstreifenmarkierungserkennung ist beispielsweise ein Kantendetektor hilfreich, der Bildbereiche mit starken Gradienten findet (vgl. Abbildung 2.3 und [Canny 1987]). Bildausschnitte, in denen diese Merkmale vorkommen, bilden eine Menge von zu validierenden Hypothesen. Die Validierung kann beispielsweise über manuell definierte Nebenbedingungen erfolgen:

So erwartet man beispielsweise bei Fahrstreifenmarkierungen aufgrund ihrer genormten Strichbreiten zwei in etwa parallel ausgerichtete Kanten, welche einen Abstand in einem bestimmten Intervall aufweisen. Meist sind

die Fahrstreifenmarkierungen auch heller als der Straßenbelag, sodass zusätzliche Randbedingungen definiert werden können, die bestimmte Gradientenausrichtungen an den beiden Kanten verlangen. Auch ist die Definition einer Mindestkantenlänge im Bildraum vorteilhaft, um Fehldetektionen weiter zu reduzieren.

Bei Objekten mit vielen komplexen und unterschiedlichen Merkmalen, wie beispielsweise bei Autos aus verschiedenen Perspektiven oder Fußgängern, ist die manuelle Definition solcher Randbedingungen sehr aufwendig. Alternativ bieten sich sog. „überwachte Lernverfahren“ zur Hypothesenvalidierung an: Hierbei werden die detektierten Objektmerkmale eines Bildausschnitts auf einen Vektor abgebildet, welcher eine Objekthypothese im Zustandsraum des verwendeten Klassifikators darstellt. Der Klassifikator wird dabei aus Lerndaten approximiert, wobei für das Anlernen neben dem Eingabevektor auch eine korrespondierende „Ground Truth“-Objektklasse benötigt wird. Für genauere Informationen zur Merkmalsgenerierung bzw. überwachten Lernverfahren sei auf [Sun 2006] bzw. Abschnitt 4.6.3 verwiesen.



Abbildung 2.4.: Optischer Fluss (rote Pfeile) für ein Verkehrsszenario, in dem das Ego-Fahrzeug einen LKW überholt (rechter Fahrstreifen) und gleichzeitig von anderen Fahrzeugen überholt wird (linker Fahrstreifen)

Eine zweite Methode zur Umgebungserfassung ist modellärmer und ermöglicht teilweise die Detektion allgemeiner Objekte. Hierzu wird der sog. „optische Fluss“ aus zwei aufeinanderfolgenden Bildern bestimmt, welcher die Verschiebung von korrespondierenden Bildpunkten mithilfe eines Vektorfeldes beschreibt. Das gemittelte Ergebnis einer Bilderserie für einen optischen Fluss ist beispielhaft in Abbildung 2.4 dargestellt.

Eine Möglichkeit diesen zu berechnen besteht in der Annahme, dass Grauwerte an korrespondierenden Stellen in den Einzelbildern eine konstante Helligkeit aufweisen. Das entsprechende Vektorfeld kann dann mithilfe von Helligkeitsdifferenzen berechnet werden. Für Details zu dieser Methode sei auf [Lucas 1985] verwiesen.

Für die Erzeugung des optischen Flusses ist es wichtig, dass bestimmte Teile und Punkte in den Bildern wiedererkannt werden können. So ist beispielsweise die Berechnung für eine einfarbige Fläche mit konstanter Helligkeit ohne eindeutige Texturen nicht möglich, wie es bei Straßenbelägen häufig der Fall ist. Auch sich wiederholende Muster sowie Belichtungsänderungen erschweren eine eindeutige Bestimmung der Verschiebung.

Das erzeugte Vektorfeld wird anschließend mithilfe weiterer Algorithmen ausgewertet: Diese nehmen an, dass der optische Fluss durch eine tatsächliche Relativbewegung von Umfeldobjekten entstanden ist und nicht durch andere Projektionseffekte beeinflusst worden ist. Zur Objektdetektion wird das Bewegungsfeld, welches sich aus Kameraparametern, der Egobewegung und der Annahme einer planaren Ebene ergibt, mit dem tatsächlichen optischen Fluss verglichen. Differenzen deuten auf Umfeldobjekte hin. Für beispielhafte Algorithmen hierzu sei auf [Giachetti 1998] verwiesen.

Zur Objektzustandsbestimmung kann die ermittelte Vektorlänge innerhalb des optischen Flusses herangezogen werden: Diese ist für eine Bildposition von Entfernung und relativer Geschwindigkeit abhängig. Ist Letztere aus der bekannten Ego-Geschwindigkeit und statischen Objektannahme bekannt, so lässt sich eine Tiefeninformation generieren. Je nach Kameraausrichtung und Relativbewegung können hier unterschiedlich gut ausgeprägte Detektionsbereiche entstehen.

Bei der häufig anzutreffenden frontalen Kameraausrichtung und Vorwärtsbewegung des Ego-Fahrzeuges gilt: Je größer die Entfernung eines Objektpunktes zur Bewegungsachse ist, desto genauer kann die Tiefeninformation bestimmt werden. Dies ist durch die stärkere Verschiebung von Bildpunkten in diesen Bereichen begründet, was Störungen durch Nicken und Wanken des Ego-Fahrzeuges und die durch die Bilddiskretisierung verursachten Ungenauigkeiten besser ausgleicht.

Ohne Relativgeschwindigkeitsannahme kann die Entfernung eines Objekts zwar nicht modellarm bestimmt werden, jedoch deren Quotient [Meyer 1992]. Dieser beschreibt die Zeit bis zur Kollision (engl. „Time-To-Collision (TTC)“) mit dem Ego-Fahrzeug bzw. bei ausreichendem lateralen Versatz die Zeit bis zum Passieren des Objektes. Diese Eigenschaft wird abgewandelt im Abschnitt 4.4.3 wieder aufgegriffen.

Bei der Bewertung von Monokamerasensoren fällt insbesondere die Lü-

cke zwischen theoretisch erreichbarer Leistung und Stand der Technik auf: Einerseits werten diese Sensoren den für die menschliche Wahrnehmung wichtigen visuellen Informationskanal aus und erhalten somit fast alle Informationen, wie sie auch ein Mensch bekommen würde. Da die Verkehrsinfrastruktur besonders für den visuellen Wahrnehmungskanal optimiert ist, haben diese Sensoren theoretisch das Potenzial, alle wesentlichen Umfeldaspekte detektieren zu können.

Andererseits ist jedoch die sichere Wahrnehmung von Entitäten mit komplexen und variablen Merkmalen problematisch. Diese erfolgt häufig über mustererkennende Algorithmen, welche auf spezielle Klassen trainiert sind und meist keine perfekte Klassifikationsleistung liefern, was beispielsweise in [Schiele 2009] im Bereich der Fußgängererkennung gezeigt wird. Auch erzeugen diese Klassifikatoren keine Entfernungsinformation, sodass diese über weitere Modellannahmen generiert werden muss. Hierzu kann der Vergleich von Abbildgröße und angenommener realer Größe herangezogen werden. Dennoch bleibt die Gewinnung eines genauen Objektzustandsvektors mit Monokamerasensoren sehr schwierig, was eine Relevanzklassifikation von Objekten erschwert.

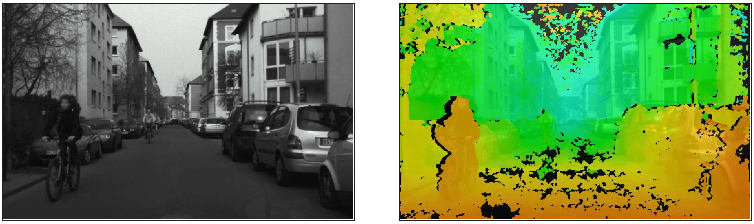
Als passiver Sensor ist die Monokamera zudem von nicht steuerbaren Lichtverhältnissen abhängig, sodass die Detektionsbedingungen in der Umwelt sehr variabel ausfallen können, was eine korrekte Klassifikation weiter erschweren kann.

Neben der Mustererkennung wurde noch die modellarme Möglichkeit vorgestellt, die Objektdetektion mithilfe des optischen Flusses zu bewerkstelligen. Diese Methode liefert jedoch bei der üblichen frontalen Kameraausrichtung für Objekte im Bereich des Fahrschlauchkorridors aufgrund der geringen Punktverschiebungen im Bild eine schlechte Detektionsleistung, wobei jedoch für viele Fahrerassistenzapplikationen gerade dieser Bereich wichtig ist. Des Weiteren ist bei dieser Methode meist die Auswertung von mehreren Bildern notwendig, was eine gewisse Verzögerung in erkennbaren Gefahrensituationen bedeuten kann.

In der Vergangenheit wurden Monokameras deshalb vor allem zur Detektion von Entitäten mit klaren oder gar genormten Merkmalen verwendet, wie Verkehrszeichen und Fahrstreifenmarkierungen. Bei diesen sind bis auf den lateralen Versatz und fahrstreifenrelativer Gierwinkel kaum weitere Zustandsinformationen für die Fahrerassistenzapplikationen notwendig. Obwohl diese Objekte von den Mustern her recht einfach sind, gibt es auch hier Fehlalarme und ausbleibende Erkennung durch verschmutzte oder teilweise verdeckte Verkehrszeichen bzw. durch Teernähte oder durch abgefahrene Fahrbahnmarkierungen.

2.2.2. Stereokamerasensor

Stereokamerasensoren bestehen aus zwei Kameras, welche in einem bestimmten Abstand (Basisbreite) voneinander angeordnet sind und einen Umgebungsausschnitt aus unterschiedlichen Perspektiven vermessen. Die Messungen erfolgen synchron, wobei in den beiden Abbildern eine Tiefenschätzung durch Vergleich der Verschiebung (Disparität) einzelner Bildpunkte bzw. Muster erzeugt wird. Die Tiefe ist antiproportional zur Disparität und proportional zur Basisbreite und Brennweite der Objektive. Durch die Synchronität der beiden Messungen wird die Auswertung kaum durch Nick- und Wankbewegungen des Ego-Fahrzeuges oder unbekannten Objektdynamiken gestört, wie es beim optischen Fluss der Fall ist. Für eine genauere Beschreibung der geometrischen Grundlagen, die für die Verarbeitung von Stereobildern notwendig sind, sei auf [Faugeras 2004] verwiesen.



(a) Originalbild (b) Disparitätsbild

Abbildung 2.5.: Beispielhafte Disparitätsberechnung (rechtes Bild) für eine Verkehrsszene (linkes Bild). Korrespondierende Punkte sind je nach Entfernung orange (nah) oder blau (fern) eingefärbt. Für schwarze Pixel konnte keine Entfernungsinformation generiert werden.

Bei der Bilderauswertung wird versucht, eine möglichst dichte Tiefenkarte zu erstellen. Wie beim optischen Fluss ist die Disparitätsbestimmung für monotone Flächen und sich wiederholende Muster jedoch nicht immer eindeutig möglich, sodass die Karte eventuell Löcher unbekannter Disparität aufweist, was Abbildung 2.5 beispielhaft zeigt. Um Objektinformationen aus der generierten Tiefenkarte zu erzeugen, wird zunächst eine Ebenenschätzung erzeugt. Hierzu wird in der Tiefenkarte ein kontinuierlicher Gradient in Höhenrichtung gesucht, der durch die Fahrbahnebene verur-

sacht wurde. Tiefenschätzungen, die von der Ebenenschätzung abweichen, werden einem Clusteralgorithmus unterworfen und zu einzelnen Objekten zusammengefasst. Häufig wird hierbei das Quadermodell verwendet, sodass die Objekthypothese eine Längen-, Breiten- und Höhenschätzung aufweist. Bei der Clusterung von Objekten mit monotonen Flächen, für die keine vollständige Disparitätsinformationen generiert wurden, können Ungenauigkeiten und Mehrdeutigkeiten auftreten: Eine genaue Konturschätzung für solche nur teilweise erfassten Objekte ist schwierig und im Extremfall können solche Objekte sogar in mehrere Objekthypothesen zerfallen.

Stereokamerasensoren können zusätzlich Algorithmen zur Auswertung von Einzelbildern implementieren und erreichen somit mindestens die Leistungsfähigkeit einer Monokamera. Durch die Ebenen- und Tiefenschätzung können sogar mehr Randbedingungen in den Klassifikatoren verwendet werden, sodass diese genauer ausfallen. So können beispielsweise detektierte Kanten als Fahrstreifenmarkierung ausscheiden, wenn diese nicht die Höhe der Ebenenschätzung aufweisen. Auch können die meisten runden Verkehrszeichen von Rädern anhand der Höhenschätzung unterschieden werden.

Zur Objektdetektion verwenden Stereokamerasensoren jedoch die modellärmere und robustere Wahrnehmung über die Disparitätskarte, anstatt einen Klassifikator zu verbessern. Die Objektklassifikation kann jedoch auch sinnvoll in der Stereokamera eingesetzt werden: Einerseits können allgemeine aus der Disparitätskarte gewonnene Umfeldobjekte einer Klasse zugeordnet werden, die ein Fahrerassistenzsystem unterschiedlich behandeln kann, andererseits können Monoalgorithmen bei gleicher Auflösung höhere Reichweiten erzielen. Letzteres ist damit begründet, dass die Disparitätsbestimmung auf Pixelebene ab einer bestimmten Entfernung spätestens an der endlichen Bildauflösung scheitert, während Objekte ab einer bestimmten Größe noch als Muster erkennbar sind.

Insgesamt können Stereokamerasensoren wie Monokamerasensoren viele Umfeldentitäten erfassen, besitzen aber zusätzlich eine bessere Objekterkennung und durch eine Kombination von Mono- und Stereokameraalgorithmen eine vergleichsweise gute Zustandsschätzung. Durch die Bestimmung einer Tiefenkarte können zudem Klassifikatoren für andere Umfeldentitäten bessere Randbedingungen nutzen, sodass diese genauer ausfallen. Durch die Ebenenbestimmung können zudem Nick- und Wankbewegungen des Ego-Fahrzeuges geschätzt werden, welche u. a. zur Messwertkompensation anderer Sensoren verwendet werden können.

Nachteilig bei aktuellen Stereokamerasensoren ist insbesondere die begrenzte Reichweite der Disparitätskarte, was eine robuste Objekterken-

nung nur in einem mittleren Abstand ermöglicht. Eine Reichweitensteigerung ist aufwendig, da die benötigte Bildauflösung quadratisch zur Entfernung ansteigt. Auch erfordert eine Vergrößerung der Auflösung eine entsprechend verbesserte Optik und eine genauere Kalibrierung. Eine Erhöhung der Basisbreite oder Brennweite würde zwar die erzielbare Reichweite vergrößern, jedoch gleichzeitig den auswertbaren Öffnungswinkel des Sensors reduzieren.

Als passiver Sensor ist die Stereokamera wie die Monokamera auch von externen Lichtverhältnissen abhängig. So müssen detektierbare Entitäten ein Mindestmaß an Ausleuchtung aufweisen, was insbesondere bei Dämmerung oder Dunkelheit problematisch sein kann: Im schlechtesten Fall muss hier die Ausleuchtung alleine von den Scheinwerfern des Ego-Fahrzeuges vorgenommen werden, wobei die Lichtintensität mit steigender Entfernung abnimmt und somit den Detektionsbereich potentiell weiter einschränkt. Auch starke externe Lichtquellen wie Scheinwerfer können die Messergebnisse von schwach beleuchteten Objekten negativ beeinflussen.

Als weiterer Nachteil ist die schlechtere Verbaubarkeit zu nennen, da der Sensor so in ein Fahrzeug integriert werden muss, dass seine Sicht möglichst vor Schmutz geschützt wird. Hierzu kommt wie bei der Monokamera als kostengünstige Lösung praktisch nur der Bereich hinter dem Innenspiegel infrage, welcher durch den bereits integrierten Scheibenwischer schmutzfrei gehalten werden kann und die freie Sicht des Fahrers nicht behindert. In diesem Bereich sind insbesondere kompakte Sensoren gefragt, wobei die Breite einer Stereokamera jedoch prinzipbedingt mindestens so groß sein muss wie dessen Basisbreite. Die Basisbreite kann man ohne Reichweitenverlust jedoch nicht reduzieren, ohne dass man die Bildauflösung wieder entsprechend erhöht.

2.2.3. Radarsensor

Radarsensoren emittieren elektromagnetische Wellen, welche von Objekten teilweise reflektiert werden. Diese reflektierte Strahlung kann vom Radarsensor detektiert werden, wobei sich abhängig vom Verfahren folgende Messgrößen direkt gewinnen lassen:

- Frequenz bzw. Phasenlage
- Intensität
- Signallaufzeit (nur bei gepulsten Modulationen)

Die unterschiedlichen Verfahren differieren in der Modulation der emittierten Strahlen, welche sich durch eine Zeit-Frequenz-Funktion darstellen lässt.

Im Automotivbereich werden meist kontinuierliche Modulationsverfahren eingesetzt, welche sich durch eine gleichmäßige und geringe Sendeleistung sowie eine geringere Störanfälligkeit gegenüber Umwelteinflüssen auszeichnen. Hierbei wird die Signallaufzeitmessung nicht mehr direkt ausgewertet, sondern indirekt über Frequenzdifferenzen.

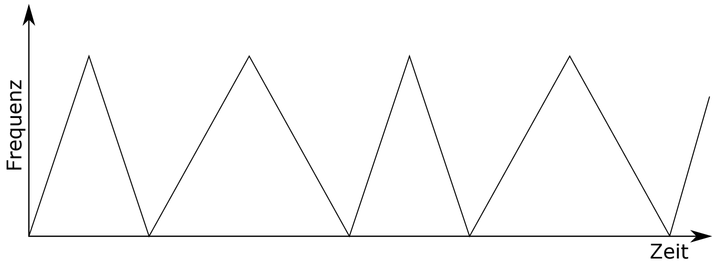


Abbildung 2.6.: Zeit-Frequenzdiagramm für eine bespielhafte Dauerstrich-Frequenzmodulation

Beispielhaft soll dies am Dauerstrich-Frequenzmodulation-Verfahren (engl. „Frequency Modulated Continuous Wave (FMCW)“) dargestellt werden, welches auch exemplarisch in [Winner 2009a] vorgestellt wird. Hierbei wird die Sendefrequenz über die Zeit linear erhöht oder erniedrigt, wobei ein Frequenzbereich nicht verlassen wird. Die Zeit-Frequenz-Funktion wird somit über eine Folge von Rampen gebildet. Ein Beispiel dazu ist in Abbildung 2.6 dargestellt.

Ein Objekt, welches sich mit einer bestimmten radialen Relativgeschwindigkeit bewegt, reflektiert diese Signale und verschiebt deren Trägerfrequenz des ausgesandten Signals durch den Dopplereffekt. Der Empfänger ermittelt die Frequenzverschiebung der aktuell gesendeten Rampe mit der empfangenen, welche im Wesentlichen aus einer Linearkombination zwischen relativer Objektgeschwindigkeit und Entfernung resultiert. Um den Abstand und die Geschwindigkeit eines einzelnen Objektes aufzulösen, ist die Auswertung zweier unterschiedlicher Rampen erforderlich, was in Abbildung 2.7 dargestellt ist.

Alternativ könnte man Geschwindigkeit und Position auch aufgrund der zeitlichen Verschiebung ermitteln. Dieses Vorgehen erfordert jedoch die exakte Bestimmung der Rampenstarts oder -endes, worauf zugunsten der

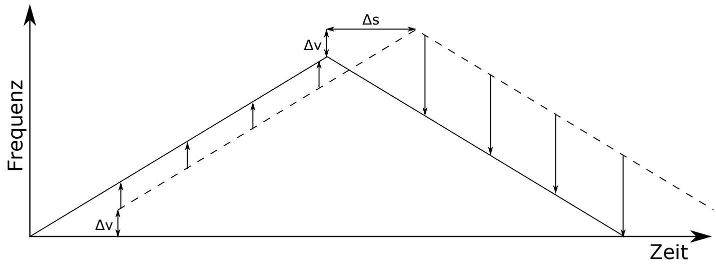


Abbildung 2.7.: Bei der FMCW-Auswertung werden meist Differenzfrequenzen (Pfeile) ausgenutzt, welche sich über einen Großteil der Rampe recht exakt mitteln lässt. Die Erkennung der Signallaufzeit Δs bzw. der Dopplerverschiebung Δv erfordert hingegen die punktuelle Erkennung des Starts oder Endes einer Rampe (Doppelpfeile).

Robustheit gegenüber Störungen und einer vereinfachten hardwaretechnischen Umsetzung verzichtet wird.

Schwieriger ist die Auswertung, wenn sich mehrere Objekte im Umfeld befinden: In diesem Fall wird eine gesendete Rampe von vielen Umfeldobjekten reflektiert, sodass mehrere Lösungen existieren können. Durch Auswertung weiterer Rampen können häufig jedoch eindeutige Lösungen erzeugt werden. Nichtsdestotrotz können ungünstige Objektkonstellationen hier Scheinziele verursachen.

Um diese möglichst zu vermeiden, verwenden praktisch alle Radarsensoren interne Trackingalgorithmen, welche die neu generierten Objekthypothesen mit vorher bekannten abgleichen. Diese Maßnahme verringert die Scheinzielrate, kann aber für neu detektierte Objekte eine Verzögerung bedeuten, bis diese als existent erkannt werden. Für Algorithmen zur Berechnung solcher Existenzwahrscheinlichkeiten in Trackingalgorithmen sei auf Abschnitt 4.4 verwiesen.

Die vorgestellte Auswertung erlaubt nur radiale Entfernungen zu bestimmen: Um eine genaue Position zu ermitteln und Mehrdeutigkeiten zu vermeiden, ist zusätzlich eine Azimutwinkelschätzung notwendig. Auch hier gibt es verschiedene Verfahren, welche beispielsweise in [Winner 2009a] vorgestellt werden. Sie basieren darauf, bekannte Geometrien, Phasendifferenzen und Feldstärken der generierten Strahlenkeulen bei der Auswertung der Rückstrahlintensitäten zu berücksichtigen. Im Gegensatz zu Lidarsensoren lassen sich die Strahlenkeulen von Radarsensoren aufgrund der

größeren Wellenlänge weniger fein fokussieren, sodass dieses Verfahren eine im Allgemeinen schlechtere Objektdiskriminierung über Azimutwinkel erlaubt.

Besonders schwierig für Radarsensoren ist deshalb die Erkennung sog. „Gassensituationen“, in denen zwei Objekte mit annähernd der gleichen Geschwindigkeit nebeneinander fahren. Diese Objekte weisen einen annähernd gleichen radialen Abstand und Geschwindigkeit auf, weshalb sie nur über den Azimutwinkel unterschieden werden können. Ist die Auflösung zu grob, können diese beiden interferierenden Objekte als ein einzelnes großes aufgefasst werden, wobei sich scheinbar die Lücke zwischen den beiden Objekten schließt. Dieser Fall kann insbesondere auch bei statischer Randbebauung auftreten, da diese exakt die gleiche relative Objektgeschwindigkeit aufweisen.

Die vorgestellte Messwertauswertung basiert insgesamt darauf, dass jedes relevante Objekt möglichst in seinem definierten Ursprung oder gleichmäßig über seine gesamte Fläche einen gewissen Anteil der Radarwellen zum Sensor reflektiert.

Einerseits kann je nach Geometrie und Ausrichtung des Objektes der Radarquerschnitt variieren und möglicherweise so wenig Signalleistung zurückreflektiert werden, dass diese unterhalb der Detektionsschwelle liegt: So beschreiben auch die ISO-Normen für ACC [ISO 2002, 2008], dass Radarsensoren nur Objekte in einem definierten Bereich detektieren müssen, welche einen Radarquerschnitt von mindestens 10 $\text{m}^2 \pm 3\ \text{m}^2$ aufweisen. Dies entspricht lt. den ISO-Normen nur 95 % aller Fahrzeuge, sodass für diese keine sichere Detektion gefordert wird.

Andererseits können an einem ausgedehnten Objekt bestimmte Bereiche die Radarwellen deutlich besser zurückreflektieren als andere: So werden in [Winner 2009a] die Einstiegstreppen von Lkws genannt, welche häufig einen erhöhten Radarquerschnitt aufweisen. Aber auch andere konkave Flächen, wie beispielsweise Radkästen und Gehäuse von konventionellen Rücklichtern können bevorzugte Reflexionsflächen darstellen. Diese befinden sich häufig an Ecken der Objektkontur, sodass trotz einer geanuen Azimutwinkelauflösung die Interpretation schwierig bleibt, wie eine Objektkontur unbekannten Ausmaßes an diesen ausgewerteten Punkt anzuordnen ist. Erkennt der Radarsensor mehrere Reflexionspunkte der Objektkontur, so kann die Konturschätzung mithilfe von Clusteralgorithmen verbessert werden (vgl. [Maurer 2004]).

Neben Objekten, die einen schlechten Radarquerschnitt aufweisen, existieren auch Scheinobjekte im Sinne der Fahrerassistenzapplikation mit zum Teil hohen Radarquerschnitten. Hierbei sind einerseits überfahrbare Objekte wie beispielsweise Getränkedosen [Jordan 2006; Steinmeyer

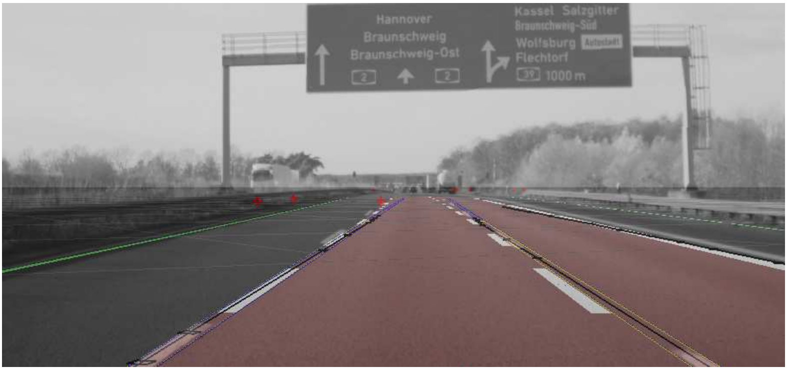


Abbildung 2.8.: Systematisches, durch Tracking aufrechterhaltenes Radar-Fehlziel (rotes Kreuz, mitte) verursacht durch Schilderbrücke

2009] oder Bahnschienen zu nennen, andererseits auch unterfahrbare Objekte wie eine in Abbildung 2.8 dargestellte Schilderbrücke.

Um die Probleme mit statischen Scheinzielen zu vermeiden, blenden viele Radarsensoren statische Objekte aus bzw. solche, bei denen während des Trackings keine Bewegung festgestellt werden konnte. Dieses Vorgehen wird auch von den ISO-Normen [ISO 2002, 2008] für ACC unterstützt.

Andere Radarsensoren versuchen Scheinziele herauszufiltern, indem sie Rückstrahlintensitäten und Winkelmessungen über eine längere Strecke beobachten. So werden in [Jordan 2006] Gassensituationen erkannt, indem im Trackingalgorithmus eine hohe Varianz der Messwerte zur Objekthypothese festgestellt wird, was durch einzelne Winkelmessfehler interferierender Objekte verursacht wird. Über- und unterfahrbare Objekte werden hingegen über eine Verteilungsanalyse der Rückstrahlintensitäten klassifiziert, welche sich von relevanten Hindernissen unterscheidet. Die unterschiedlichen Verteilungen resultieren aus einer unterschiedlich starken Änderung des Elevationsmesswinkels über die Strecke bei über- oder unterfahrbaren Objekten im Vergleich zu relevanten Hindernissen auf Höhe des Radarsensors. Problematisch an der Methode ist jedoch die Notwendigkeit der längeren Objektbeobachtung, was im Vergleich zur Sensorreichweite eine späte Hindernisklassifikation zur Folge haben kann. Auch können vom Klassifikator relevante Ziele als irrelevant angesehen werden (vgl. [Jordan 2006]).

2.2.4. Lidarsensor

Lidarsensoren basieren auf der Aussendung eines Lichtimpulses, welche von einem oder mehreren Objekten reflektiert wird. Die Laufzeit des reflektierten Lichtimpulses misst der Lidarsensor und ermittelt daraus die Objektentfernung. Die meisten Lidarsensoren sind mehrzielfähig und somit in der Lage, mehrere Reflexionen eines Lichtimpulses auszuwerten.

Aufgrund der geringen Wellenlänge von Infrarotlicht lassen sich diese Lichtstrahlen besser fokussieren als dieses bei Radarsensoren der Fall ist. Vorteilig an diesem Messprinzip ist die Möglichkeit, genaue Objektkonturen vermessen zu können, nachteilig die teilweise hohe Empfindlichkeit gegenüber Nick- und Wankbewegungen des Ego-Fahrzeuges. Dies kann den Messpunkt beim Zielobjekt leicht verschieben, was bei Objekten mit nicht senkrecht zum Sensor ausgerichteten Oberflächen zu geringen Abstandsänderungen führen kann. Da die Dynamikeigenschaften von keinem Lidarsensor direkt gemessen, sondern über Trackingverfahren abgeleitet werden, können solche Effekte Scheinbeschleunigen hervorrufen.

Für den Automotivbereich existieren verschiedene Lidarsensoren, welche sich vor allem in der Methodik zur Abtastung der Umgebung und der Anzahl der Scan-Ebenen unterscheiden:

- **Feste Lidarsensoren:** Diese Lidarsensoren enthalten ein Array von Sende- und Empfangseinheiten, welche jeweils einen konstanten Bereich abdecken. Sie sind sehr robust, da sie keine bewegliche Mechanik enthalten.

- **Schwenkende Lidarsensoren:** Diese Lidarsensoren weisen eine bewegliche Optik auf, mit der die Strahlen zu einem gewissen Grad gelenkt werden können. Sie erlauben es somit einen größeren Bereich mit weniger Sende- und Empfangseinheiten abzudecken, als feste Lidarsensoren.

- **Laserscanner:** Bei Laserscannern rotiert die Ausrichtung der Sende- und Empfangseinheit. Dies kann direkt durch eine Rotation des gesamten Sensors geschehen oder indirekt über einen rotierenden Spiegel, der die emittierten und reflektierten Laserstrahlen in die entsprechende Richtung bzw. Empfangseinheit lenkt. Laserscanner weisen hohe Öffnungswinkel von bis zu 360° auf und werden häufig im Forschungsbereich eingesetzt.

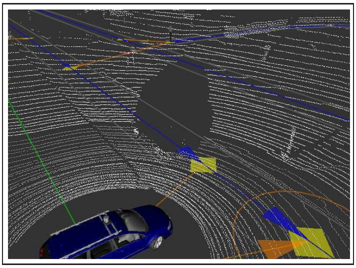
Schwenkende Lidarsensoren und Laserscanner haben den Nachteil, dass zur Synchronisation eines Messzyklus eine Eigenbewegungsschätzung notwendig ist, da die einzelnen Messungen sich über einen gewissen

Messzeitraum verteilen. Ungenaue Eigenbewegungsschätzungen können somit Schereffekte hervorrufen.

Alle Lidarsensoren benötigen freie Sicht auf ein Objekt, um dieses zu detektieren. Einschränkungen durch Nebel, starker Niederschlag oder Verschmutzungen am Sensor können Detektionen von Objekten verhindern. Allerdings sind die meisten Sensoren selbstdiagnosefähig, sodass solche Störungen erkannt und an die Fahrerassistenzapplikation gemeldet werden können (vgl. [Geduld 2009]).



(a) Schwarze Limousinen sind für Lidarsensoren potenziell Ziele mit geringer Reflektivität



(b) Im Laserscan können kaum Reflexionen (weiße Punkte) für dieses Ziel detektiert werden

Abbildung 2.9.: Nichtdetektiertes Ziel eines Laserscanners (Bilder aus [Petrovskaya 2009])

Wie bei Radarsensoren hängt die Detektierbarkeit anderer Objekte von der Reflektivität bzgl. der emittierten Strahlung ab, weshalb die Geometrie und Ausrichtung des Objektes auch hier ein wesentlicher Faktor darstellt. Als weiterer Einflussfaktor kommt bei Lidarsensoren jedoch eine hohe Lichtabsorbtionsfähigkeit von einigen Objekten hinzu: Während die meisten Fahrzeuge für Radarsensoren viele gut reflektierende Metalloberflächen aufweisen, können Autos mit schwarzen Oberflächen nicht immer detektiert werden (vgl. Abbildung 2.9).

Neben Objekten können bei entsprechender Sensorausrichtung auch Fahrstreifenmarkierungen detektiert werden, indem neben Signallaufzeit zusätzlich die Intensität der Reflexion gemessen wird (vgl. [Dietmayer 2005b]). Hierbei wird der Umstand ausgenutzt, dass Fahrbahnmarkierungen speziell darauf optimiert sind, möglichst viel Licht zurückzureflektieren, damit sie der Fahrer gut wahrnehmen kann. Vorteilhaft gegenüber der kamerabasierten Fahrstreifenerkennung ist das aktive Messprinzip, sodass

die Erkennung weitestgehend unabhängig von den Lichtverhältnissen der Umgebung ist.

2.2.5. Ultraschallsensoren

Ultraschallsensoren nutzen sog. „piezoelektrische Materialien“, welche beim Anlegen einer Spannung Schallwellen emittieren können und umgekehrt eine Spannung erzeugen, wenn sie durch reflektierte Schallwellen verformt werden.

Das Messprinzip zur Objekterkennung beruht insgesamt auf einer Laufzeitmessung der emittierten und reflektierten Schallwellen, wobei im Gegensatz zu Lidarsensoren statt der Lichtgeschwindigkeit nur die viel langsamere Schallgeschwindigkeit in der Luft berücksichtigt werden muss, was geringere Anforderungen an den Zeitgeber in der Signalverarbeitung nach sich zieht. Als Ergebnis liefern Ultraschallsensoren radiale Abstände von Objekten im Messsektor oder sie lassen auf einen leeren Sektor bei nicht detektierter Reflexion schließen. Für Genaueres zum piezoelektrischen Effekt und dessen Anwendung in Ultraschallsensoren sei auf [Noll 2009] verwiesen.

Ultraschallsensoren haben eine vergleichsweise kurze Reichweite, sind dafür aber relativ kostengünstig und kompakt, weshalb sie vielfach im Bereich der Parkassistenzsysteme verwendet werden. Um ausgedehnte Objekte und Objektpositionen genauer bestimmen zu können, werden mehrere mit bekanntem Abstand angeordnete Ultraschallsensoren verwendet, welche die unterschiedlichen Laufzeiten eines einzelnen Sendeimpulses messen und durch eine Multilateration eine bessere Schätzung erreichen.

Neben Objektabstandsbestimmungen verwenden einige Fahrerassistenzsysteme die inverse Belegungsinformation und validieren damit, ob ein bestimmter Bereich frei oder belegt ist. Anwendungen hierbei sind die Parklückenvermessung und Spurwechselassistenz. Bei Letzterer wird aufgrund der geringen Reichweite nur der tote Winkel überwacht, sodass vor schnell überholenden Fahrzeugen nicht rechtzeitig gewarnt werden kann.

Ultraschallsensoren haben wie alle aktiven Sensoren den Nachteil, dass ihre emittierten Signale ausreichend zum Empfänger zurückreflektiert werden müssen. Diese Eigenschaft kann bei einigen Objekten eingeschränkt sein, wie beispielsweise bei Metallketten, welche Parktaschen begrenzen. Auch müssen diese Sensoren verstärkt mit Fremdsignalen umgehen können, da andere dynamische Objekte und insbesondere das Ego-Fahrzeug selbst ständig präsente Störschallquellen darstellen, wobei unterschiedlichste Frequenzen auftreten können.

2.2.6. Bewertung

Zur Umfeldwahrnehmung werden technologisch unterschiedliche Umfeldsensoren eingesetzt, wobei jede Sensortechnologie ihre Stärken und Schwächen aufweist. Exemplarisch wird dies hier für die Zustandsbestimmung für Umfeldobjekte dargestellt, da viele Fahrerassistenzsysteme auf dessen genaue Schätzung angewiesen sind: So kann beispielsweise eine ungenaue Objektpositionsschätzung die Ursache dafür sein, dass ein Spurwechselassistent fälschlicherweise annimmt, dass sich ein Objekt im Nachbarfahrstreifen befindet und unnötig warnt.

Die Sensoren unterscheiden sich einerseits in der maximalen Reichweite, in der sie typische Umfeldobjekte detektieren können, andererseits haben sie aufgrund ihres Messprinzips unterschiedliche Genauigkeiten in der Zustandsbestimmung von detektierten Objekten. So können bestimmte Radarsensoren hohe Reichweiten von 200 m oder mehr erzielen. Ebenso erreichen viele Lidarsensoren diese Reichweite bei gut reflektierenden Zielen.

Bei der Genauigkeit der Zustandsbestimmung ist hingegen keine allgemeingültige Aussage möglich: Diese hängt stark von einzelnen Zustandsattributen ab. So erlauben Kameras aufgrund ihrer hohen Auflösung eine sehr gute Winkelschätzung, ebenso wie gut fokussierbare Lidarsensoren. Dadurch kann ebenfalls eine Objektkontur geschätzt werden, wobei dies bei der Monokamera aufgrund der fehlenden Tiefeninformation nur ungenau mit Hilfe von Modellannahmen gelingt.

Eine Geschwindigkeitsschätzung hingegen gelingt diesen Sensoren nur über die zeitliche Ableitung der Positionsschätzung, sodass diese ungenauer ist als bei Radarsensoren. Insbesondere Monokamerasensoren weisen eine nur ungenaue Entfernungsschätzung auf, sodass die longitudinale Komponente eines Geschwindigkeitsvektors mit größerer Unsicherheit behaftet ist.

Für die praktische Nutzung der Sensoren für Fahrerassistenzsysteme ist neben deren Leistungsfähigkeit auch eine leichte Verbaubarkeit und eine hohe Robustheit gegenüber Umwelteinflüssen wichtig. So sollten Sensoren möglichst kompakt und an universell vielen Orten im Fahrzeug verbaut werden können und gleichzeitig möglichst unempfindlich gegenüber Regen, Schnee, Dunkelheit und Verschmutzungen sein. Hier sind die aktuell häufig genutzten Radar- und Ultraschallsensoren im Vorteil, da diese nahezu unsichtbar hinter durchlässigen Kunststoffen versteckt bzw. entsprechend lackiert werden können. Kamera- und Lidarsensoren hingegen benötigen freie Sicht, welche auch nicht durch Verschmutzungen beeinträchtigt werden sollte, weshalb häufig nur Verbauorte hinter Scheiben mit

	Reichweite	Entfernungsschätzung	Winkelschätzung	Geschwindigkeitsschätzung	Konturschätzung	Umfeldentitäten	Umwelteinflüsse	Verbaubarkeit
Monokamera	+[1]	-	++	- -	o	+	o	o
Stereokamera[2]	o[1]	+	++	o	++	++	o	o/-[3]
Radar	++[4]	++	+	++[5]	-	-	++	+
Lidar	++	++	+	+	+	o	+	o
Ultraschall	-	+	o	- -	-	-	+	++

Tabelle 2.2.: Bewertung der Sensortechnologien

Wischanlage infrage kommen. Kamerasensoren sind als passive Sensoren zusätzlich empfindlich gegenüber einer schlecht ausgeleuchteten Umgebung. Eine Übersicht der vorgestellten Sensoren und deren Bewertung ist in Tabelle 2.2 dargestellt. Weitere Bewertungen einzelner Sensortechnologien sind in [Rasshofer 2005] zu finden.

Neben der genauen Schätzung des Objektzustandes ist für Fahrerassistenzsysteme ebenso wichtig, dass der Sensor alle relevanten Ziele möglichst zuverlässig detektiert. Scheinziele („Falsch positive Objekte“) als auch nicht erkannte relevante Ziele („Falsch negative Objekte“) können sich negativ auf die Qualität der Umfeldwahrnehmung und somit negativ auf das gesamte Fahrerassistenzsystem auswirken: Wird beispielsweise eine Messung für ein ACC-System so missinterpretiert, dass im Fahrschlauch ein Objekt fälschlicherweise wahrgenommen wird, so erfolgt eine

[1] Die Reichweite von Kamerasensoren kann mithilfe entsprechender Objektive auf Kosten des verfügbaren Öffnungswinkels in erhöht werden. Hierbei ist die Fahrzeugeigenbewegung zu beachten, welche den auswertbaren Bildbereich in dynamischen Situationen einschränken kann. Hier sind typische Konfigurationen bewertet.

[2] In Stereokameras können auch Monobildverarbeitungsalgorithmen eingesetzt werden, sodass sich diese nach erschöpfter Stereoreichweite wie eine Monokamera verhält.

[3] Die Verbaubarkeit hängt wesentlich von der verwendeten Basisbreite ab

[4] Hohe Reichweiten erzielen einige Radarsensoren, welche im 77 GHz-Bereich arbeiten.

[5] Radarsensoren können sehr gut relative Radialgeschwindigkeiten messen. Um einen kartesischen Geschwindigkeitsvektor zu erhalten, muss zusätzlich die Winkeländerung betrachtet werden.

unnötige Bremsung. Übersieht ein ACC-System ein relevantes Objekt im Fahrschlauch, dann unterbleibt eine notwendige Bremsung.

Sensor	**Ursachen für falsch positive Objekte**	**Ursachen für falsch negative Objekte**
Monokamerasensor	Objekthypothesen, welche ähnliche Muster aufweisen wie Objekte im Sinne des Klassifikators, aber keine Objekte in dem Sinne sind	Schlecht ausgeleuchtete Objekte; Objekte, die nicht dem Muster der vorgegebenen Objektmenge entsprechen
Stereokamerasensor	Mehrdeutigkeiten in der Disparitätsberechnung durch sich wiederholende Muster	Objekte mit geringer Disparität (homogene Flächen), schlecht ausgeleuchtete Objekte, Objekte mit geringer Höhe (keine Separierung durch Ebene)
Radarsensor	Über- und unterfahrbare metallische Objekte (Schilderbrücken, Getränkedosen), Mehrdeutigkeiten in der Signalauswertung (z. B. durch Gassensituationen)	Objekte mit geringem Radarquerschnitt
Lidarsensor	Fahrbahn oder Schilderbrücke bei größerer, nicht detektierte Nickbewegung des Ego-Fahrzeuges oder bei größeren Steigungsänderungen der Fahrbahn	Lichtabsorbierende Objekte; Objekt mit planaren Flächen, die die Lichtstrahlen nicht zum Sensor zurückreflektieren
Ultraschallsensor	Störschallquellen anderer Fahrzeuge, Interferenzen durch andere Ultraschallsensoren, Regen	Objekte, welche wenig Schallwellen reflektieren (z. B. dünne Metallkette)

Tabelle 2.3.: Mögliche Ursachen von falsch positiven oder falsch negativen Objekten bei verschiedenen Sensortechnologien

Ursache hierfür ist zum einen, dass alle Umfeldsensoren nicht exklusiv die Zustandsgrößen von relevanten Objekten direkt messen, sondern diese in einer Vielzahl von Einzelmesswerten mehr oder weniger verborgen sind. So misst eine Schwarz/Weiß-Kamera nur Helligkeitswerte auf einem CCD-Chip. Erst bildverarbeitende Algorithmen interpretieren die Messung,

sodass Objekte und deren Zustand wahrgenommen werden können. Aufgrund der nahezu unendlich großen Menge von möglichen Bildzuständen ist es jedoch für solche Algorithmen schwierig, aus jedem Bild immer die richtigen Objekte und deren Zustandsgrößen zu extrahieren. Ähnliches gilt für Lidar- und Radarsensoren, welche eine Vielzahl von Reflexionsinformationen korrekt interpretieren müssen.

Zum anderen können je nach Sensortechnologie bestimmte Objekte nicht oder nur sehr spät erfasst werden. So müssen Objekte ein Mindestmaß an Texturierung aufweisen, damit sie von einer Stereokamera erfasst werden können. Frühzeitig durch einen Radarsensor detektierbare Objekte müssen einen bestimmten Mindestradarquerschnitt zum Radarsensor hin besitzen. Umgekehrt gibt es Objekte, die zwar irrelevant für das Fahrerassistenzsystem sind, aber gleichzeitig sehr gut emittierte Strahlen reflektieren. So können überfahrbare Getränkedosen und Gullydeckel Ursache für Fehlziele für einen Radarsensor sein, genauso wie unterfahrbare Schilderbrücken. Eine Übersicht möglicher Ursachen für falsch positive und falsch negative Objekte ist in Tabelle 2.3 zusammengefasst.

Als weitere Fehlerquelle sind verletzte Modellannahmen zu erwähnen, welche in weiteren Verarbeitungsschritten erfolgen. Die Darstellung dieser Fehlerquellen erfolgt im folgenden Kapitel 3.

2.3. Zusammenfassung

In diesem Kapitel sind verschiedene Sensortechnologien sowohl zur Schätzung der relativen Position und Dynamik des Ego-Fahrzeuges als auch zur Detektion und Zustandsbestimmung von Umfeldobjekten beschrieben worden. Um ein genaues Umfeldmodell zu erstellen, ist sowohl die Schätzung der relativen Objektzustände als auch der Eigenbewegung notwendig, was eine Trennung der Dynamik von Ego-Fahrzeug und Umfeldobjekten erlaubt. Zusätzlich verwenden viele Fahrerassistenzsysteme die Egozustandsgrößen direkt als Eingangsgrößen in deren jeweiligen Applikation.

Bereits bei der Ego-Zustandsschätzung existieren jedoch einige sporadische und auch systematische Fehlerquellen: Neben nicht gemessenen Radschlupf, unbekannten Radumfang und Radstand bei Verwendung von Raddrehzahlsensoren, sind auch viele im Fahrzeug verwendete Gierraten- und Beschleunigungssensoren empfindlich gegenüber Vibrationen, Fahrbahnunebenheiten oder Fahrbahnneigungen. Somit kann ein aus diesen Daten gewonnener Zustandsvektor je nach Dynamik des Ego-Fahrzeuges und Fahrbahnbeschaffenheit mehr oder weniger ungenau sein. Abhilfe

könnte eine darauf spezialisierte Inertialplattform schaffen, welche um Größenordnungen genauere Messwerte liefert, jedoch ist hierbei mit deutlichen Zusatzkosten zu rechnen.

Wie bei den Odometriesensoren müssen auch bei Umfeldsensoren verschiedene mögliche Ursachen berücksichtigt werden, wenn man die Güte der Messungen bewerten möchte. Neben Ungenauigkeiten in der Zustandsbestimmung ist hier jedoch noch eine weitere Unsicherheitsquelle zu berücksichtigen: Während Odometriesensoren von der Existenz des Ego-Fahrzeuges ausgehen können, können einzelne Umfeldsensoren unter Umständen falsch positive Objekte im Sinne des Fahrerassistenzsystems liefern oder relevante Objekte nicht wahrnehmen.

Dabei ist eine heterogene Sensorkonfiguration vorteilhaft, um den Einfluss dieser Unsicherheitsquellen zu minimieren: Einerseits können bekannte Stärken von Sensoren bei der Bestimmung einzelner Zustandsattribute berücksichtigt werden, andererseits sollte es in den meisten Fällen zu widersprüchlichen Informationen zwischen einzelnen Sensoren kommen, wenn ein Sensor ein Scheinziel liefert oder ein relevantes Ziel übersieht. Herausfordernd sind hierbei die genaue Zustandsschätzung und die Auflösung auftretender Widersprüche zu einem statistisch genauen Ergebnis, wobei möglichst alle verfügbaren Messungen, deren Abhängigkeiten und andere Informationen berücksichtigt werden sollten.

Um dieses zu erreichen, wird im folgenden Kapitel eine Sensordatenfusionmodul vorgestellt, welches ein umfangreiches Umfeldmodell in Form verschiedener Teilschätzer beinhaltet. Diese Teilschätzer werden in darauf folgenden Kapiteln dann so kombiniert, dass eine breite Bewertungsgrundlage für daraus aufbauende Schätzalgorithmen gegeben ist. Letztere werden ebenfalls in den beiden Kapiteln 4 und 5 dargestellt.

Kapitel 3.

Sensordatenfusion

3.1. Einleitung

Für die Umsetzung Fahrerassistenzsystemen ist eine Umfeldwahrnehmung elementare Voraussetzung: Wie im vorherigen Kapitel vorgestellt, werden zu diesem Zweck Sensoren eingesetzt, die bestimmte Aspekte des Umfeldes messen können. Aus diesen Messdaten werden die Hypothesen erzeugt, welche validiert und anschließend an die Fahrerassistenzapplikationen weitergeleitet werden müssen. Hierzu existieren im Wesentlichen zwei unterschiedliche Konzepte, wobei aber auch Hybridformen und verteilte Systeme denkbar sind:

Im ersten Konzept wird jeweils ein Steuergerät für ein Assistenzsystem verwendet, an dem exklusiv die relevanten Sensoren zur Umfeldwahrnehmung und entsprechende Aktoren bzw. Informationssysteme angebunden sind. Deren Umfeldmodell ist soweit optimiert, dass ausschließlich die für das einzelne Fahrerassistenzsystem relevanten Größen bestimmt werden. Häufig ist dieses Steuergerät direkt im Sensor zu finden, sodass dieses zusätzlich die Signalvorverarbeitung bewältigt. Vorteil dieses Konzeptes ist die Unabhängigkeit der Systeme untereinander. Durch die konzeptionell bedingte Modularität arbeitet beispielsweise ein ACC-System unabhängig von einem Fahrstreifenverlassenswarner, sodass diese auch separat verbaut und verkauft werden können. Nachteilig an diesem Konzept ist, dass vorhandene Informationen nicht vollständig genutzt werden. So könnte beispielsweise die Fahrstreifenschätzung hilfreich für eine Fahrschlauchgenerierung sein, sodass die Zielauswahl für ACC-Systeme verbessert wird.

Ein zweites Konzept sieht dagegen ein zentrales Umfeldmodell für verschiedene Fahrerassistenzsysteme vor, welches diverse Aspekte des Umfeldes enthält. Dieses Umfeldmodell wird meist in einem zentralen Softwaremodul implementiert, welches auf der einen Seite die Sensordaten von allen relevanten Umfeld- und Fahrzeugsensoren verarbeitet und auf der anderen Seite das Umfeldmodul bereitstellt. Aus dieser Software können die benötigten Informationen für verschiedene Fahrerassistenzsys-

teme extrahiert werden. Dieses Softwaremodul heißt Sensordatenfusion (SDF). Vorteilhaft an diesem Konzept ist vor allem die Möglichkeit, mehr Informationen aggregieren zu können, sodass das Umfeldmodell eine potenziell höhere Qualität und einen größeren Umfang aufweist.

Insgesamt kann solch Sensordatenfusion je nach Anzahl und Granularität der integrierten Modelle ein sehr komplexes System darstellen und somit einen beträchtlichen Entwicklungsaufwand nach sich ziehen. Um die Komplexität zu beherrschen, erfolgt eine funktionale Dekomposition in Einzelkomponenten, welche so verknüpft sind, dass sie definierten Hierarchien und Anordnungen unterliegen. Diese Gesamtstruktur bezeichnet man als zugrunde liegende Architektur des Systems. Großen Einfluss auf die Komplexität und somit auf die Systemarchitektur haben die Anforderungen an eine Sensordatenfusion, welche sich aus den umzusetzenden Fahrerassistenzsystemen ergeben.

In diesem Kapitel werden typische Anforderungen an eine Sensordatenfusion untersucht, existierende Ansätze vorgestellt und eine eigene Sensordatenfusionsarchitektur dargestellt. Diese bildet die Basis für die folgenden Kapitel, in denen bestimmte Schätzer genauer untersucht werden.

3.2. Anforderungen

Die Anforderungen an eine Software haben großen Einfluss auf dessen Architektur und die zu integrierenden Algorithmen. Sie vollständig vorher zu erfassen, erleichtert die Entwicklung, da nachträgliche Anforderungen mitunter große Änderungen an der Softwarearchitektur und damit an Zeit und Kosten verursachen können.

Für eine Sensordatenfusion für Fahrerassistenzapplikationen sind folgende allgemeine Anforderungen sinnvoll:

- **Genau**: Die verwendeten Modelle sollten in der Lage sein, die Zustandsgrößen der Umfeldschätzung in einer Güte zu generieren, die für das Fahrerassistenzsystem ausreichend sind[1].
- **Robust**: Die Sensordatenfusion sollte gegenüber Störungen möglichst unempfindlich sein, wie z. B. ein kurzzeitig ausgefallener Sensor oder bei einzelnen Fehlmessungen.
- **Probabilistisch**: Die Umfeldmodellabstraktion und die Imperfektion der Sensordaten schränken die Genauigkeit der Umfeldschätzung

[1]Eine formalere Definition des Genauigkeitsbegriffes ist beispielsweise in [Brahmi 2013] zu finden

ein. Dessen Unsicherheit sollte quantitativ mitgeschätzt werden, beispielsweise in Form von Varianzen und Existenwahrscheinlichkeiten[2].

- **Echtzeitfähig**: Fahrerassistenzapplikationen sollten mit einer definierten maximalen Latenz alle notwendigen Informationen über ein hinreichend aktuelles Umfeldmodell bei einer bestimmten Zielplattform erhalten[3]

- **Modular**: Neue Sensoren oder neue Modellaspekte sollten sich mit möglichst wenig Aufwand in die Sensordatenfusion integrieren lassen.

Die genannten Anforderungen lassen sich zum Teil nicht unabhängig betrachten. So hängt es dann von der Applikation ab, welche Anforderungen einen hohen und welche einen niedrigeren Stellenwert besitzen, was unterschiedliche Architekturen zur Folge haben kann.

So vereinfacht sich beispielsweise die Umsetzung der Genauigkeits- und Echtzeitanforderung, wenn sog. „synchrone Sensoren" eingesetzt werden vgl. [Darms 2007, S. 20]. Diese messen gleichzeitig, was zeitliche Prädiktionen reduziert und Retrodiktionen ausschließt. Eine Synchronitätsanforderung an Sensoren schränkt allerdings die Modularität ein, da die Sensoren dann eine gleiche Zyklusrate aufweisen und sich an einem einheitlichen Synchronisationssignal orientieren müssen. Die Messungen „asynchroner Sensoren" können hingegen in beliebiger Reihenfolge mit unterschiedlicher Latenz die Sensordatenfusion erreichen. Hier ist eine Erweiterung der Sensordatenfusion in Form eines Messwertepufferspeichers oder retrodiktionsfähiger Algorithmen notwendig (vgl. [Stüker 2004, S. 63 ff] und [Muntzinger 2011]). Diese können aber Echtzeitfähigkeit und die Genauigkeit der Schätzung einschränken.

Weiteren Einfluss auf die Architektur und Algorithmen hat der Ort der Signalvorverarbeitung. Sensoren enthalten häufig Steuergeräte, die aus den Rohdaten Merkmalsdaten extrahieren oder sogar ein internes Tracking besitzen. Je stärker die Sensorsteuergeräte die Messdaten vorverarbeiten, desto geringer ist der Aufwand der Sensordatenfusion. Dies kann die Entwicklung echtzeitfähiger Algorithmen erleichtern, schränkt allerdings die Genauigkeit der Umfeldschätzung ein: Je stärker die Sensoren Messdaten vorverarbeiten, desto höher ist der Informationsverlust (vgl. [Darms 2007, S. 18 f]).

[2] Eine formale Definition von probabilistischen und statistisch korrekten Schätzern ist im Anhang A.1.2 zu finden

[3] Eine Definition zu harter Echtzeitfähigkeit ist im Anhang A.1.1 dokumentiert.

Auch die Auswahl der Schätzalgorithmen für das Umfeldmodell muss sorgfältig aufgrund der Anforderungen erfolgen. Hierbei ist die Genauigkeitsanforderung eines Fahrerassistenzsystems mit der Echtzeitanforderung bzgl. einer Zielplattform auszubalanzieren. Im folgenden Abschnitt wird beispielsweise das sog. „Kalmanfilter“ als dynamischer Zustandsschätzer vorgestellt, zu dem es auch ein Monte-Carlo-Pendant gibt, das „Partikelfilter“ heißt. Letzteres kann bei idealer Modellierung eine höhere Genauigkeit liefern, ist jedoch meist viel speicher- und rechenaufwendiger.

3.3. Dynamische Zustandsschätzer

Um die Anforderung der genauen und robusten Umfeldmodellierung Rechnung zu tragen, verwenden praktisch alle Sensordatenfusionen sog. „Dynamische Zustandsschätzer“. Diese bilden eine Algorithmenklasse, welche Messreihen von einem oder mehreren Sensoren nutzen, um bestimmte Aspekte eines sich dynamisch ändernden Umfeldes zu approximieren. Die meisten Zustandsschätzer liefern zusätzlich eine Schätzgüte in Form einer Wahrscheinlichkeitsdichtefunktion. Die Varianz dieser Verteilung repräsentiert die Unsicherheit dieser Schätzung.

Dynamische Zustandsschätzer sind modellbasiert: Sie enthalten u. a. ein Dynamikmodell, mit dem ein Zustand beispielsweise in der Zeitdomäne vorhergesagt werden kann. Auch die Unsicherheiten des Dynamikmodells und der Messungen werden meist in Form von System- bzw. Messrauschen modelliert.

Eine häufig genutzter dynamische Zustandsschätzer ist das „Kalmanfilter“, dessen Algorithmus im Anhang A.2 vorgestellt wird. Dieses Filter bildet meist einen guten Kompromiss aus Echtzeitfähigkeit und Genauigkeit, weshalb es auch in vielen existierenden Sensordatenfusionsarchitekturen Verwendung findet, welche im folgenden Abschnitt 3.4 vorgestellt werden. Das Kalmanfilter bildet auch eine wichtige Grundlage für die im Abschnitt 3.5 entwickelte Sensordatenfusionsarchitektur.

3.4. Stand der Technik: Sensordatenfusionsarchitekturen

Sensordatenfusionsarchitekturen können nach verschiedenen Kriterien klassifiziert werden, wobei Art (synchrone oder asynchrone Sensoren) und Sichtbereiche (komplementär, konkurrierend oder kooperativ) Unterscheidungsmerkmale darstellen können. Auch können zentrale Sensordatenfu-

sionsmodule von verteilten Architekturen unterschieden werden. Für eine Übersicht möglicher Architekturen sei auf [Dietmayer 2005a] verwiesen.

Bisher wurden diverse Sensordatenfusionen im automotiven Bereich für unterschiedlichste Projekte und Anforderungen entwickelt. Allerdings findet man in der Literatur wenig über komplexe Sensordatenfusionsarchitekturen, welche sehr viele Umfeldaspekte gleichzeitig berücksichtigen und kombinieren. Häufig wird ausschließlich das Tracking dynamischer Umfeldobjekte mit dem Kalmanfilter beschrieben (siehe beispielsweise [Stüker 2004; Darms 2007]).

Im Zusammenhang mit Wettbewerben der DARPA[4] sind Sensordatenfusionen entstanden, die neben dem Objekttracking auch eine Überfahrbarkeitsschätzung mittels Belegungsgitter enthalten [Effertz 2009; Thrun 2006; Urmson 2008; Rojo 2007; Kammel 2008]. Dies ist damit begründet, dass diese Wettbewerbe in komplexen[5] Umgebungen wie Wüste oder urbanen Bereich stattfanden und somit modellarme Umgebungsrepräsentationen vorteilhaft waren. Insbesondere bei der „Urban Challenge“ war auch mit anderen sich bewegenden Verkehrsteilnehmern zu rechnen, sodass ein zusätzliches klassisches Objekttracking Erfolg versprechend war.

Viele Arbeiten enthalten zwar eine Fahrbahnschätzung und kombinieren diese mit anderen Teilfusionen. Diese Schätzung basiert bei vielen Architekturen jedoch auf einer hochgenauen Ortung in Kombination mit genauem digitalem Kartenmaterial. Letzteres ist entweder a priori bekannt (vgl. Projekt „iCar“ in [Weiser 2009]) oder wird vorher aufgrund von detektierten Begrenzungen generiert (vgl. Projekt „GTI 53+1“ in [Kompaß 2008]). Auch bei den Fusionen der „Urban Challenge“-Teilnehmer spielten Fahrbahnverlaufsschätzungen mittels lokaler Messdaten eine eher untergeordnete Rolle: So wird beispielsweise in [Rojo 2007] beschrieben, dass aufgrund der hohen Dichte von vorgegebenen GPS-Wegpunkten diese in der Wettbewerbsdomäne unnötig seien und die Fahrbahnmarkierungsinformationen hauptsächlich für eine laterale fahrbahnrelative Positionsschätzung des Ego-Fahrzeuges verwendet wurden. Auch in [Kammel 2008] wird ein ähnlicher Ansatz verfolgt, welcher aus akkumulierten Fahrbahnmarkierungen und Bordsteinen den Versatz zum gegebenen Kartenmaterial schätzt.

Einige andere Fusionen integrieren die lokal ermittelten Fahrbahndaten in bestimmte Teilschätzer: So werden in [Kolski 2006] die detektierten Fahrstreifenmarkierungen in ein Belegungsgitter mit geringer Gewichtung

---

[4] Die „Defense Advanced Research Projects Agency“ (DARPA) ist eine Technologieabteilung des US-Verteidigungsministerium. Sie hat verschiedene Wettbewerbe veranstaltet, in denen Fahrzeuge bestimmte Strecken autonom und auf Zeit zurücklegen mussten. Diese befanden sich in der Mohave-Wüste (Grand Challenge) bzw. im städtischen Umfeld (Urban Challenge).

[5] Komplexe Umgebungen lassen sich nicht mit Modellen beschreiben, die nur wenige freie Parameter aufweisen.

eingetragen, sodass diese bevorzugt nicht überfahren werden. Diese modellarme Repräsentation ermöglicht die Einbringung von Fahrbahnmarkierungen beliebiger Geometrie, hat jedoch den Nachteil, dass aufgrund des fehlenden Fahrbahnmodells weder eine Assoziation von Messungen noch eine Prädiktion des Fahrbahnverlaufes möglich ist. Dies mach den Schätzer empfindlich gegenüber Messrauschen, fehlinterpretierten Messungen und fehlenden Messungen.

Wenige andere Arbeiten integrieren eine lokale modellbehaftete Fahrbahnschätzung: So ist im Rahmen des Teilprojektes SAFELANE des EU-Förderprojektes PReVENT die Fusionierung von Fahrbahn- und Objektdaten untersucht worden (s. [Ahrholdt 2006; Polychronopoulos 2005b,a]).

Dabei wird einerseits die Fahrbahnschätzung durch die gemessenen Trajektorien vorausfahrender Fahrzeuge verbessert, andererseits ermöglicht die Fahrbahnschätzung selbst eine bessere Fahrstreifenzuordnung anderer Verkehrsteilnehmer. Als weitere Informationsquelle zur Fahrbahnschätzung dienen neben detektierten Fahrstreifenmarkierungen auch digitale Kartendaten und die Krümmung der Egotrajektorie. Letztere Datenquelle bietet sich nur an, wenn die Fahrzeugquerregelung nicht auf den fusionierten Fahrbahndaten basiert, da ansonsten Rückkopplungen denkbar sind.

Als Fahrbahnmodell wird das sog. „Klothoidenmodell“ genutzt, welches als Basisparameter durch Krümmung und Krümmungsänderung spezifiziert ist. Um es zum Ego-Fahrzeug korrekt anzuordnen, ist zusätzlich die Angabe der Drehung und Ablage notwendig. Als dynamischer Zustandsschätzer finden mehrere erweiterte Kalmanfilter Anwendung, wobei sie die Klothoide mithilfe eines Polynom dritten Grades approximieren. Auf eine direkte Verwendung des Klothoidenmodells wird verzichtet, da dieses ungünstige mathematische Eigenschaften aufweist[6].

Die Kalmanfilter schätzen für jede einzelne Informationsquelle mit Ausnahme der digitalen Karte die Krümmung und Krümmungsänderung separat. Die Ergebnisse werden über eine dynamische Gewichtung zu einer Klothoide fusioniert. Diese Gewichtung richtet sich u. a. nach Varianz der Schätzung, Entfernung und Verfügbarkeit der Informationsquelle. Eine Übersicht dazu ist in Abbildung 3.1 dargestellt.

Neben den in SAFELANE verwendeten Datenquellen wird in [Schuberth 2010] vorgeschlagen, zusätzlich statische Randbebauungen zu berücksichtigen. Diese seien häufig parallel zur Fahrbahn angeordnet, sodass sie dessen Verlauf widerspiegeln. Als Beispiel werden Bäume, Leitplan-

[6] Eine allgemeine Klothoide kann nicht durch eine Funktion beschrieben werden, sondern nur durch eine Kurve.

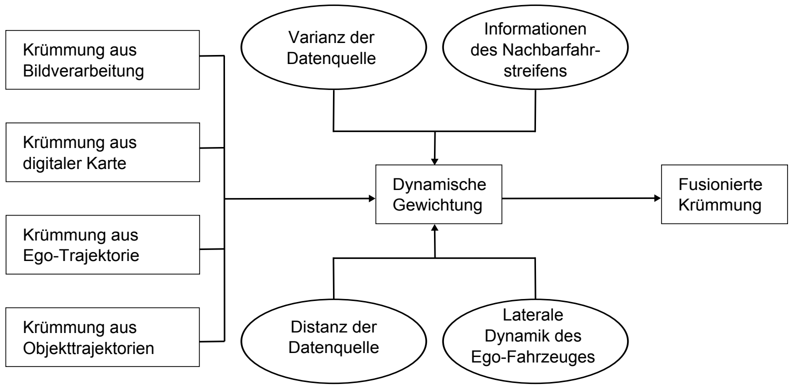


Abbildung 3.1.: Fahrbahnschätzung für SAFELANE (vgl. [Polychronopoulos 2005b])

ken und Leitpfosten genannt. Im Gegensatz zu SAFELANE werden die Informationsquellen in [Schuberth 2010] nicht durch Kalmanfilter, sondern anhand selbstdefinierter Gütemaße gewichtet.

Alle vorgestellten Fusionen zeichnen sich dadurch aus, dass sie eine Odometrieschätzung vornehmen. Diese wird häufig durch eine separate hochgenaue Inertialplattform durchgeführt, welche meist mit einer hochgenauen Ortung mittels „Differential GPS" kombiniert wird. Solche fertigen Plattformen verwendeten praktisch alle DARPA-Wettbewerber (vgl. [Thrun 2006; Effertz 2009; Rojo 2007; Urmson 2008]) und andere Versuchsträger, die auf genauem Kartenmaterial (vgl. [Weiser 2009]) basieren. Diese Plattformen zeichnen sich durch einen sehr geringen Fehler in der Egozustandsbestimmung aus, sind aber aus Kostengründen in der hohen Qualität für einen massenhaften Einsatz in Serienfahrzeugen heute ungeeignet.

Im Rahmen industrienaher Arbeiten wird deshalb vermehrt auf bereits im Fahrzeug bestehende Odometriesensoren zurückgegriffen (s. [Mählisch 2009; Kolski 2006] und PREVenT-Projekt). Bei diesen ist zusätzlich die Entwicklung eines auf die Fahrzeugsensorik abgestimmten Schätzers erforderlich. Herausfordernd ist der Umgang mit der geringeren Schätzgüte, da dieser Schätzfehler auf viele andere Teilfusionen propagiert wird.

Im Folgenden soll eine Sensordatenfusion entwickelt werden, welche ein möglichst umfangreiches Umfeldmodell liefert und dabei vielversprechende Aspekte vorhandener Ansätze kombiniert. Die verwendeten Sensoren sollen möglichst seriennah sein: So soll die Sensordatenfusion bei-

spielsweise auf eine hochgenaue Ortung und Inertialplattform verzichten und auch kein hochgenaues Kartenmaterial verwenden. Stattdessen sollen heute übliche Kartendaten mit begrenzter Genauigkeit genutzt werden, welche heutige Navigationssysteme verwenden. Gleiches gilt für Umfeldsensoren, welche einerseits möglichst kostengünstig sein müssen und andererseits unauffällig im Auto verbaut werden können.

3.5. Entwicklung einer Sensordatenfusionsarchitektur

Im vorherigen Abschnitt sind einige Sensordatenfusionen vorgestellt worden, welche sich auf unterschiedliche Aspekte konzentrieren. Im Hinblick auf eine umfangreiche Umfeldrepräsentation wird in diesem Abschnitt eine Sensordatenfusion vorgestellt, welche einige dieser Ansätze aufgreift, diese kombiniert und erweitert[7].

Die Sensordatenfusion soll dabei in einzelne Schätzer unterteilt werden, die folgende Umfelddaten bereitstellen:

- **Egofusion**: Eigenbewegungsschätzung
- **Objektfusion**: Verfolgung und Zustandsschätzung von Umfeldobjekten
- **Belegungsgitterfusion**: Befahrbarkeitsschätzung
- **Fahrbahnfusion**: Fahrbahnverlaufsschätzung
- **Digitale Karte**: Straßenvorausschau

Hierbei sollen die genannten Anforderungen beachtet werden. Zunächst werden jedoch die verwendeten Architekturmuster beschrieben, die hilfreich sind, die Modularitätsanforderung zu bewältigen.

3.5.1. Pipe & Filters-Architektur

Als dominierendes Architekturmuster wurde für diese Sensordatenfusion die sog. „Pipe & Filter-Architektur“ gewählt, in der Teilkomponenten durch sog. „Filter“ dargestellt werden. Diese sind über sog. „Pipes“ miteinander verbunden, über welche Daten ausgetauscht werden. Ein Beispielgraph ist in Abbildung 3.2 dargestellt.

[7] Die Schätzer dieser Sensordatenfusion wurden von einer Gruppe von Mitarbeiter aus dem Konzernverbund der Volkswagen AG entwickelt. Der Autor gehört dieser Entwicklergruppe an und hat mit Ausnahme der Egofusion an allen Modulen mitgewirkt. Der Relevanzschätzer zur Objektfusion (vgl. Kapitel 4) und das Belegungsgittermodul (vgl. Kapitel 5) und hat er eigenständig konzeptioniert und implementiert.



Abbildung 3.2.: Filtergraphbeispiel mit 4 Instanzen von 3 Filtern A, B und C. Filterinstanzen haben links Eingangspins und rechts Ausgangspins

Ein Filter hat verschiedene Eingangsempfangs- und Ausgangssendestellen, in denen ein Datenstrom entgegengenommen, verarbeitet, gegebenenfalls zwischengespeichert und versendet wird. Diese Stellen werden im verwendeten Framework (s. [Schabenberger 2007]) als „Pins“ bezeichnet.

Die Filter werden in einer Filtergraphkonfiguration instanziiert und miteinander verbunden. Dieser sog. „Filtergraph“ stellt einen gerichteten Graphen dar, wobei die Knoten die Filterinstanzen und die Kanten die Pipes sind. Die Erstellung und die äußere Ablaufsteuerung des Graphens werden dabei durch einen sog. „Filtergraphmanager“ aufgrund einer vom Benutzer erstellten Konfiguration vorgenommen.

Die Pipe & Filter-Architektur erlaubt die lose Kopplung verschiedener Komponenten. Diese müssen weder die Funktion noch die interne Struktur von Nachbarfiltern kennen. Die gemeinsame Schnittstelle ist über die Datenstruktur definiert, welche dem zu empfangenden Datenstrom zugrunde liegt.

Pipe & Filter-Architekturen unterstützten die Parallelisierung von Software: So können unterschiedliche Zweige im gerichteten Graphen parallel auf unterschiedlichen Prozessoren berechnet werden und somit bestimmte Echtzeitanforderungen leichter erfüllen. Dieses wird in Abschnitt 3.5.8 genauer behandelt.

3.5.2. Verwendetes Schichtenmodell

In der entwickelten Sensordatenfusion existieren über 90 verschiedene Filter, welche miteinander interagieren, weshalb mögliche Filtergraphen dementsprechend komplex aufgebaut sein können. Um eine weitere Ordnung in die Architektur zu bringen, ist fast jedes Filter einer sog. „Softwareschicht“ zugeordnet. Eine Softwareschicht umfasst eine grundlegendere Zuständigkeit als ein einzelnes Filter und hat maximal zwei Nachbarschichten, mit denen sie kommuniziert.

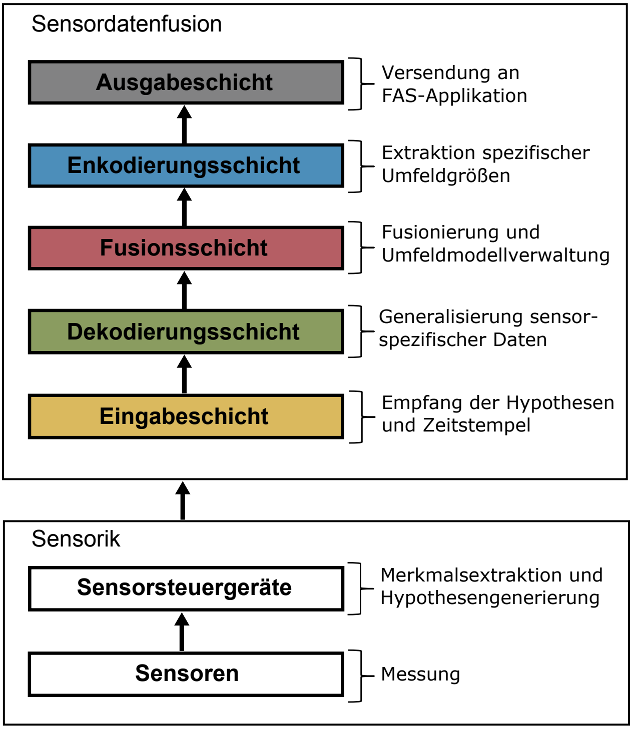


Abbildung 3.3.: Verwendetes Schichtenmodell

In der Sensordatenfusionsarchitektur existieren insgesamt 5 Schichten: Eine Eingabe-, eine Dekodierungs-, eine Fusions-, eine Enkodierungs- und eine Ausgabeschicht, wie sie in Abbildung 3.3 dargestellt sind. Diese haben folgende Aufgaben:

Die Eingabeschicht abstrahiert den Empfang von Messdaten aus verschiedensten Quellen. Die empfangenen Daten werden in der Dekodierungsschicht in einheitliche Formate konvertiert und somit von sensorspezifischen Protokollen abstrahiert.

Die einheitlichen Formate werden dann in einer Fusionsschicht genutzt, um ein virtuelles Umfeld aufzubauen. Hier sind einzelne Teilfusionen dafür zuständig, bestimmte Umgebungsmerkmale zu approximieren. Die Egofusion schätzt dabei die Eigenbewegung, die Objektfusion die Position und die Geschwindigkeit anderer Verkehrsteilnehmer, die Fahrbahnfusion den

Schicht	Abstraktion	Zugehörige Filter
Eingabe-schicht	Empfang	Filter zum Empfang von CAN-Daten über CAN-Bus, Empfang von UDP-Paketen über Ethernet (für Simulation) und Abspielen von Messdateien
Dekodierungs-schicht	Sensor-spezifische Formate	Filter zum Konvertieren in ein generisches Objektformat, ein generisches Fahrbahnformat oder generisches Freibereichsformat
Fusions-schicht	Umfeld	Filter zur Schätzung der Odometrie, anderer Objekte, der Fahrbahn, der Umgebung und der Kartenposition
Enkodierungs-schicht	Applikations-spezifische Formate	Filter zur Erzeugung von Ausgabeprotokollen für Ego-Fzg.-Zustand, Umfeldobjekten, Größen für Längs- und Querregelung
Ausgabe-schicht	Senden	Filter zum Versenden der Ausgabeprotokolle als CAN-Daten über CAN-Bus, in UDP über Ethernet oder in eine Messdatei

Tabelle 3.1.: Übersicht der Schichten in der Sensordatenfusion

Verlauf der Straße, die Belegungsgitterfusion die Befahrbarkeit der nahen Umgebung und die Digitale Karte liefert zusammen mit der Ortung bestimmte Straßenattribute. Die einzelnen Teilfusionen werden im Abschnitt 3.5.5 genauer vorgestellt. Die fusionierten Umgebungsdaten fließen anschließend in die Enkodierungsschicht.

Die Enkodierungsschicht sammelt Daten der Fusion und konvertiert diese in spezifische Formate, die für die jeweilige Fahrerassistenzapplikation geeignet sind. Beispielhaft werden vier unterschiedliche Ausgabeprotokolle generiert, welche unterschiedliche Aspekte von Fahrerassistenzsystemen abdecken. Diese werden später im folgenden Abschnitt 3.5.6 näher beschrieben.

Die Ausgabeschicht ähnelt der Eingabeschicht: Hier wird der Sendeweg des Ausgabeprotokolls abstrahiert, welches in Abschnitt 3.5.7 genauer dargestellt wird. Eine Gesamtübersicht aller Schichten, Abstraktionsaufgaben und enthaltenen Filter ist in Tabelle 3.1 zu finden.

Die meisten Filter sind den genannten Schichten zugeordnet. Es gibt jedoch einige wenige, die außerhalb dieses Schichtenmodells liegen und

mit Filtern agieren, die in verschiedenen Schichten anzutreffen sind. Dazu gehört beispielsweise der sog. „Scheduler“, der die Ablaufsteuerung der Fusion maßgeblich regelt und im Abschnitt 3.5.8 dokumentiert ist. Daneben existieren Visualisierungs- und Debuggingkomponenten, die Informationen an vielen Stellen schichtenübergreifend darstellen können. Als Beispiel sei eine 3D-Visualisierung genannt, die sowohl Messdaten im generischen Format (Schicht 2), fusionierte Daten (Schicht 3) und Daten des Ausgabeprotokolls (Schicht 4) anzeigen kann.

3.5.3. Eingabeschicht: Abstraktion der Empfangswege

Die Eingabeschicht abstrahiert den Empfang von Messdaten. Diese können über einen CAN-Bus empfangen werden, wie es im Fahrzeug oder bei Hardware-in-the-loop-Simulationen der Fall ist. In einer Offlinekonfiguration können auch Messdateien abgespielt werden, welche dann in das Framework eingespeist werden. Für Simulationszwecke ist die Kommunikation mit anderen Programmen sinnvoll. Hierbei wird das verbindungslose UDP-Protokoll über Ethernet genutzt, welches einzelne Messdatenpakete in UDP-Paketen kapselt.

Die Eingabeschicht ist zusätzlich verantwortlich für die Bereitstellung der Zeitquelle: Während die Messdaten vom CAN-Bus ihren Zeitstempel durch einen Zeitgeber auf der Empfangshardware erhalten, enthalten die aufgenommenen Messdateien und die simulierten Daten bereits definierte Zeitstempel. Diese wurden während der Messdatenaufnahme bzw. durch die Simulation festgelegt. In diesen beiden Fällen kann die Sensordatenfusion somit auch schneller oder langsamer als in Realzeit laufen. Die so generierte Zeitquelle ist Basis für die später in Abschnitt 3.5.8 vorgestellte Ablaufsteuerung der Sensordatenfusion.

3.5.4. Dekodierungsschicht: Abstraktion durch generische Formate

Da jeder Sensor in seinem spezifischen Format seine Messdaten sendet, sind Anpassungen der Sensordatenfusion an neue Sensoren notwendig. Um Adaptionen in der Fusionsschicht möglichst zu vermeiden und somit den Aufwand zu minimieren, werden im Folgenden verschiedene generische Formate definiert, welche die wesentlichsten Aspekte der Messdaten im Sinne der Sensordatenfusion vereinheitlichen und kapseln. Hierbei wird davon ausgegangen, dass bereits eine Signalvorverarbeitung stattgefun-

den hat oder dass diese in dem Dekodierungsfilter erfolgt, sodass tiefere Schichten keine Rohdaten verarbeiten müssen.

Exemplarisch werden hierzu Formate für Objektinformationen, Freibereiche und Fahrbahninformationen vorgestellt. Diese ermöglichen nicht nur einheitliche und definierte Schnittstellen für einzelne Zustandsschätzer, sondern auch für generische Visualisierungs- und Protokollierungskomponenten.

Nachteilig an diesen Formaten ist jedoch, dass nicht immer alle Messattribute in das jeweilige generische Format abgebildet werden können und somit ein Informationsverlust eintritt. Die Abbildung solcher Informationen ist jedoch nur sinnvoll, wenn sie später in den Zustandsschätzern auch Verwendung findet. Hierzu ist dann zusätzlich eine Erweiterung der Fusionsschicht notwendig.

Alle Messdaten werden von den Dekodierungsfiltern vom Sensorkoordinatensystem in das Fahrzeugkoordinatensystem transformiert, wobei bekannte Einbaupositionen und Ausrichtungen berücksichtigt werden. Außerdem erhalten diese einen aus der Zeitquelle abgeleiteten Zeitstempel, welcher den Messzeitpunkt repräsentieren soll. Hierbei werden bekannte Verzögerungen durch Signalvorverarbeitungen im Sensor oder Synchronisationsdaten beachtet.

Zusätzlich ist in allen generischen Formaten ein Sensoridentifier vorgesehen, sodass diese jederzeit einem Sensor wieder zugeordnet werden können.

Generische Objektdaten

Bei der Interpretation von Messdaten werden verschiedene Objekthypothesendarstellungen abhängig vom Messprinzip verwendet. So erzeugen beispielsweise viele Radarsensoren punktförmige Objekthypothesen mit Geschwindigkeitsinformationen, während Laserscanner und Stereokamerasensoren keine Geschwindigkeiten messen, dafür aber häufig zusätzliche Konturinformationen liefern können. Ultraschallsensoren hingegen liefern nur radiale Entfernungsinformationen, was durch ein Kreissegment dargestellt werden kann.

Um möglichst viele Attribute kapseln zu können, beschreibt das generische Format für jedes Objekt nicht nur einen umfangreichen Zustandsvektor, sondern auch, ob ein Attribut im Zustandsvektor gültig ist. Nachfolgende Module müssen vor der Verarbeitung zunächst prüfen, ob die notwendigen Attribute auch im Objekt enthalten sind.

Für jede gültige Zustandsgröße kann auch deren Unsicherheit in Form einer Varianz abgelegt werden, welche beispielsweise mithilfe eines in-

versen Sensormodells[8] und des Zustandsvektors festgelegt werden kann. Wie beim Zustandsvektor auch gibt es für jedes Element im Varianzvektor ein Gültigkeitsattribut, sodass dieses Element nicht zwangsweise generiert werden muss.

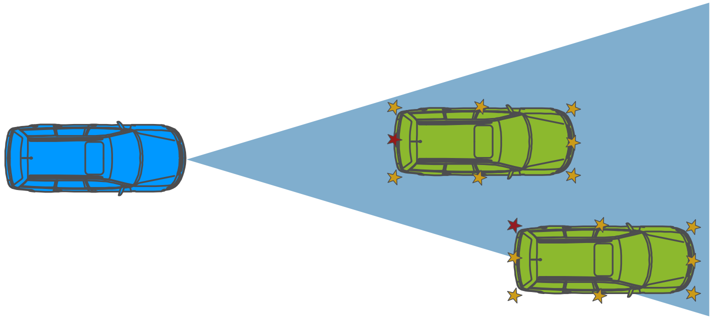


Abbildung 3.4.: Die Position jeder Objekthypothese (grüne Fahrzeuge) wird durch einen (roter Stern) von acht möglichen Referenzpunkten (alle Sterne) angegeben. Die Festlegung des Referenzpunktes erfolgt durch ein inverses Sensormodell, welches den Messvektor berücksichtigt.

Bei einigen Messungen kann der Fall auftreten, dass Objekte teilweise verdeckt oder sich teilweise außerhalb des Messfeldes befinden. Hierzu unterstützt das Format einen variablen Referenzpunkt bzgl. eines Rechteckmodells, auf welches sich der Positionsvektor bezieht. Breiten- und Längenschätzungen sind dann als Schätzungen von diesem Referenzpunkt anzusehen. Ein Beispiel hierzu ist in Abbildung 3.4 dargestellt.

Damit das Format nicht nur Sensordaten kapseln kann, sondern auch als Ausgabeformat für einen Objekttracker verwendet werden kann, ist dieses noch um einige Attribute erweitert worden. So ist neben der Trackingdauer auch ein Flag vorgesehen, welches beschreibt, ob das Objekt im letzten Zyklus gemessen oder nur prädiziert worden ist. Ein weiteres Attribut dokumentiert, ob ein Objekt neu initialisiert worden ist oder sich bereits in der Objektliste des letzten Trackingzyklus befand.

Insgesamt lassen sich durch das Format sehr viele Objektinformationen beschreiben: So ist durch das Setzen entsprechender Attribute ein Punkt-,

[8] Ein inverses Sensormodell schließt von der Messung ausgehend auf die Umgebung, während ein Sensormodell von der Umgebung ausgehend auf die Messung schließt. Inverse Sensormodelle werden zur Sensordatenfusion benötigt, während Sensormodelle beispielsweise für Simulationsumgebungen genutzt werden.

Linien- oder Rechteckmodell möglich. Außerdem ist es möglich, die Positionsinformation entweder kartesisch oder radial anzugeben.

Die generischen Objekte werden zuletzt in einer Objektliste zusammengefasst, welche einen Messzyklus eines Sensors oder eine Umfeldobjektliste des Trackers repräsentiert. Diese Liste wird noch mit einem Mess- bzw. Schätzzeitpunkt und einem Quellidentifier angereichert.

Generische Freibereichsdaten

Zur Beschreibung von Freibereichen, die direkt oder indirekt gemessen worden sind, dient ein generisches Freibereichsformat. Zur Approximation eines solchen Bereiches eignen sich besonders konkave Polygone: Dies ist einerseits dadurch begründet, dass einige Sensoren wie beispielsweise Laserscanner in bekannten diskreten Winkelschritten Entfernungsmessungen vornehmen, welche durch ein solches Polygon exakt rekonstruiert werden können, andererseits ist es damit möglich, jede andere Freibereichsform beliebig genau anzunähern. Letzteres ist für sektorförmige Freibereichsinformationen notwendig, welche beispielsweise durch Ultraschallsensoren generiert werden können.

Um die Unsicherheit einer Messung darzustellen, kann man jedem Eckpunkt des Polygons eine Varianz zuweisen. Die entfernungsabhängige Varianz bezieht sich auf einen vorher definierten Referenzpunkt, der beispielsweise im Sensorursprung liegt. Um komplexe Evidenzberechnungen innerhalb eines Polygons zu vereinfachen, wird statt einer Normalverteilung eine Dreiecksverteilung angenommen. Die Evidenz innerhalb eines Sektors in solch einem Polygon wird durch lineare Interpolation der Varianzinformationen der Eckpunkte berechnet. Zusätzlich wird dem gesamten Freibereich noch eine Gesamtevidenz zugeordnet.

Durch dieses modellarme Format können viele Freibereiche flexibel angenähert werden. Eingeschränkt wird dieses nur durch die angenommene Dreiecksverteilung und dessen singulären Referenzpunkt. Letzteres kann beispielsweise die Freibereichsgenerierung durch ein Sensorarray einschränken, wenn man jeden Sensor nicht einzeln betrachten möchte.

Generische Fahrbahndaten

Um Information über einen Fahrbahnverlauf möglichst modellarm darzustellen, wurde für das generische Fahrbahndatenformat eine Linienzugdarstellung mithilfe von Stützpunkten gewählt. Dieses kann indirekt auch modellbehaftete Darstellungen repräsentieren, indem alle Stützpunkte beispielsweise auf einem Kreis- oder Klothoidensegment abgebildet werden.

Jeder Linienzug kann durch diverse Attribute angereichert werden: Wenn dieser Fahrbahnmarkierungen repräsentieren soll, ist es beispielsweise möglich, die Art, Breite und Farbe der Markierung anzugeben. So können weiße gestrichelte Linien beispielsweise von gelben durchgezogenen Linien unterschieden werden. Solche Informationen sind u. a. hilfreich um festzustellen, auf welchem Fahrstreifen einer mehrspurigen Straße man sich befindet oder ob ein Fahrstreifenwechsel erlaubt ist.

Neben Fahrstreifenmarkierungen können auch andere Informationsquellen dargestellt werden. So ist es möglich, den Linienzügen eine Höhe zuzuweisen, sodass beispielsweise der Verlauf einer Bordsteinkante oder einer direkt am Fahrstreifen befindlichen Leitplanke dargestellt werden kann. Diese Informationen können beispielsweise die Existenz von entsprechenden Nachbarfahrstreifen verneinen.

Wie bei dem generischen Objektformat werden die Linienzüge einer Messung zu einer Liste zusammengefasst und mit einem Messzeitpunkt versehen.

Zusammenfassung

Die generischen Formate erlauben eine Abstrahierung der sensorspezifischen Formate, was die Modularitätsanforderung berücksichtigt. Tabelle 3.2 zeigt eine Übersicht der verwendeten Formate und in welchen Teilmodulen sie verwendet werden. Sie erlauben die schnelle Integration neuer Sensoren in die Sensordatenfusion.

Format	Abstrahierung	Verwendung in
Generische Objektdaten	Objekthypothesen als Punkt-, Linien, Box- oder radiales Modell	Visualisierung, Objektfusion, Fahrbahnfusion, Belegungsgitterfusion
Generische Fahrbahndaten	Fahrbahnverlaufshypothesen als Stützpunktmodell	Visualisierung, Fahrbahnfusion
Generische Freibereichsdaten	Freibereichshypothesen als Polygonmodell	Belegungsgitterfusion

Tabelle 3.2.: Generische Datenformate und Verwendung innerhalb der Sensordatenfusion

3.5.5. Fusionschicht: Teilfusionen und Dienste

Das Umfeldmodell besteht aus mehreren unabhängigen Teilen, von denen jedes in einer eigenen Fusionskomponente geschätzt und aktualisiert wird. Jede Fusionskomponente erfüllt einen spezifischen Informationsbedarf und verwendet eine entsprechende Modellabstraktion.

Da einige Teilfusionen große Datenmengen kapseln, ist es ineffizient, diese komplett an nachfolgende Filter zu schicken. Deshalb verwenden viele Teilfusionen Dienstschnittstellen, über die andere Filter die gewünschten Daten von den Teilfusionen beziehen können. Diese enthalten Methoden, welche die gekapselten Daten aufbereiten: So bieten viele Teilfusionen beispielsweise eine Zustandsprädiktion auf andere Zeitpunkte an.

Die Dienstschnittstellen erlauben zusätzlich eine Änderung der internen Datenrepräsentation oder gar den kompletten Austausch eines Schätzers: So muss ein anderer Schätzalgorithmus nur die Dienstschnittstelle reimplementieren, um transparent eingesetzt werden zu können. Dieses wird beispielsweise zu Simulationszwecken genutzt.

Egofusion

Die Egofusion schätzt den Zustand des Ego-Fahrzeuges ausschließlich mithilfe von Messungen der Odometriesensoren. Dabei werden Messungen von den Raddrehzahlsensoren der Hinterräder und vom Gierratensensor verwendet, welche in ein erweitertes Kalmanfilter eingehen. Der geschätzte Zustandsvektor umfasst eine relative Position, die zurückgelegte Strecke, eine Geschwindigkeit, eine Beschleunigung, eine Gierrate und einen relativen Gierwinkel. Die Position und der Gierwinkel werden relativ zur Startposition bzw. Startausrichtung angegeben.

Da keine globale Positionsinformationen in die Schätzung mit eingeht, erfolgt eine rein integrierende Zustandsschätzung. Diese zeichnet sich durch eine praktisch kontinuierliche Zustandsänderung und durch einen sich integrierenden Fehler aus. Schwerpunkt dieser Zustandsschätzung ist somit die Minimierung des relativen Fehlers zwischen zwei zeitlich nahen Schätzungen ohne Rücksicht auf einen globalen Fehler zu nehmen. Abbildung 3.5 zeigt beispielhaft Trajektorien zweier Schätzer mit und ohne sich integrierenden Fehler.

Die Minimierung dieses relativen Schätzfehlers ist entscheidend für die Minimierung der Schätzfehler anderer Teilfusionen: So wird der relative Fehler auf den Dynamikzustand im Objekttracker propagiert, während Gierwinkelfehler wiederholte Zellassoziationen in der Belegungsgitterfusi-

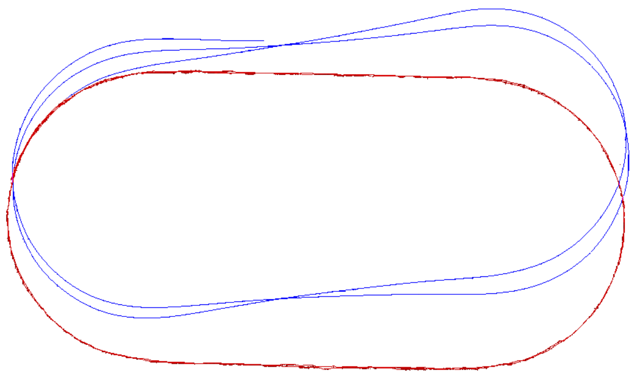


Abbildung 3.5.: Relative Odometrieschätzung (blaue Trajektorie) und globale Schätzung mittels GPS (rote Trajektorie) nach mehreren Umrundungen eines ovalen und ca. 1,7 km langen Testgeländes

on erschweren. In der Fahrbahnfusion kann eine fehlerhafte Gierrate hingegen Scheinkrümmungen verursachen.

Die Egofusion liefert eine Dienstschnittstelle, mit welcher der Egozustand eines bestimmten Zeitpunktes abgefragt werden kann. Dieses erfolgt durch Prädiktion bzw. Retrodiktion des aktuellen Zustandes, was anderen Schätzern ermöglicht, den Egozustand zu spezifischen Mess- oder Schätzzeitpunkten anzunähern. Hiervon machen alle weiteren vorgestellten Teilfusionen Gebrauch.

Objektfusion

Die Objektfusion schätzt den Zustand von Umfeldobjekten mittels generischer Objektdaten. Für die Zustandsschätzung wird ein erweitertes Kalmanfilter eingesetzt, welches neben der kartesischen Position und Objektkontur auch die Geschwindigkeit und Beschleunigung in Längs- und Querrichtung für jedes Objekt schätzt.

Als Ergebnis erhält man eine Liste von Objekthypothesen, welche jeweils aus einem Zustandsvektor und dessen korrespondierender Kovarianzmatrix besteht. Letztere dokumentiert die Unsicherheit der Zustandsschätzung, was die Bewältigung der Anforderung an eine probabilistische Sensordatenfusion unterstützt.

Im Gegensatz zur Egofusion muss die Objektfusion mehrere Zustände parallel schätzen: Hierzu ist die Assoziation einer Messung mit bereits vorhandenen Umfeldobjekten notwendig. Als Grundlage dient die sog. Mahalanobisdistanz, welche Objekthypothese (Track) und Messung vergleicht. Das Abstandsmaß gewichtet die Innovation zwischen Trackzustand und Messung $\tilde{y}_t$ (s. Gleichung A.5) mit dessen Residualkovarianz S_t (s. Gleichung A.6) im Messraum. Zusätzlich wird der Logarithmus der Residualkovarianzdeterminante als additiver Term zum Abstandsmaß hinzugefügt, um eine geringe Trackunsicherheit positiv zu gewichten (vgl. [Stüker 2004, S. 49 f]):

$$d^2(\tilde{y}_t, S_t) = \tilde{y}_t^T S_t^{-1} \tilde{y}_t + \ln(|S_t|) \tag{3.1}$$

Als Ergebnis erhält man eine Assoziationsdistanz: Je kleiner diese Distanz ist, desto geringer ist die Erwartungswertdifferenz zwischen Track und Messung gemäß der Verteilungserwartung im Messraum und desto geringer ist die Trackunsicherheit.

Diese Distanz entscheidet auch über den Lebenszyklus eines Tracks: Der Track mit der geringsten Entfernung zur Messung wird mit dieser assoziiert. Überschreitet diese Entfernung einen bestimmten Schwellwert, so wird von einem neu gemessenen Objekt ausgegangen, was eine Trackinitialisierung zur Folge hat. Kann ein Track in mehrere Zyklen mit keiner Messung assoziiert werden, so wird dieser gelöscht. Dieses kann beispielsweise auftreten, wenn ein Objekt sich aus dem Sichtbereich der Sensoren entfernt oder längerfristig verdeckt wird.

Eine Übersicht über die Trackverwaltung bietet das in Abbildung 3.6 dargestellte Flussdiagramm[9]. Der verwendete Objekttracker nutzt zusätzlich einen variablen Referenzpunkt für partielle Assoziationen und unterstützt asynchrone Sensoren, wobei zugunsten der Echtzeitfähigkeit auf einen Messdatenpuffer mithilfe retrodiktionsfähiger Algorithmen verzichtet werden konnte. Er kann auch radiale Abstandsinformationen verarbeiten, welche beispielsweise durch Ultraschallsensoren erzeugt werden. Für genauere Informationen zu Objekttrackingalgorithmen sei beispielsweise auf [Bar-Shalom 2001; Stüker 2004] verwiesen. Einen weiteren Ansatz zur partiellen Assoziation und zur Konturschätzung wird in [Ohl 2011] vorgestellt.

Neben dem ungenauen Objektzustand ergeben sich beim Objekttracking weitere Unsicherheits- und Fehlerquellen: Messungen können eventuell nicht eindeutig zu benachbarten Objekthypothesen zugeordnet werden, sodass bei Fehlassoziation eine inkorrekte Zustandsaktualisie-

[9] Durch die Rekursivität des Algorithmus ist die Terminierung von der Sensordatenmenge abhängig. Deshalb wurde auf Start- und Stoppsymbole im Flussdiagramm verzichtet.

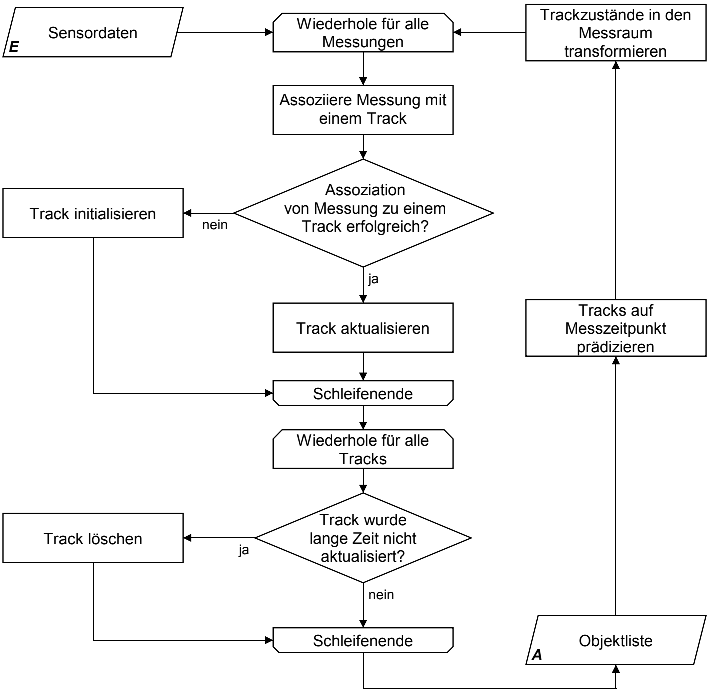


Abbildung 3.6.: Flussdiagramm des Objekttrackingalgorithmus

rung erfolgt. Bei Messausreißern kann zudem der maximale erlaubte Mahalanobisabstand überschritten werden, was die Initialisierung eines Scheinobjektes zur Folge hat. Unter ungünstigen Umständen erlauben beide Phänomene zusammen, dass ein Objekt in mehrere Objekthypothesen zerfällt.

Wie bereits in Kapitel 2 vorgestellt, können Sensoren zudem Fehlziele generieren oder relevante Ziele eventuell nicht erkennen, sodass auch hier Scheinziele initialisiert oder relevante Ziele verworfen werden.

Um diesen Unsicherheiten zu begegnen, wird eine fusionierte Relevanzwahrscheinlichkeit in einem separaten Schätzer für jedes Objekt generiert, was in Kapitel 4 ausführlich behandelt wird.

Fahrbahnfusion

Die Fahrbahnfusion schätzt den nahen Verlauf der Fahrbahn, die Anzahl der vorhandenen Fahrstreifen und deren Breite. Hierzu nutzt sie verschiedene Informationsquellen: Diese bestehen aus Fahrbahnmarkierungen in Form von generischen Fahrbahndaten, Karteninformationen über die Dienstschnittstelle der digitalen Karte und generische Objektdaten. Letztere werden daraufhin untersucht, ob sie eine Trajektorie eines dynamischen Objektes beschreiben oder aber Randbebauungen darstellen. Beides weist auf einen möglichen Fahrbahnverlauf hin. Eine beispielhafte Visualisierung dieser Informationsquellen und die resultierende Fahrbahnschätzung ist in Abbildung 3.7 darstellt.

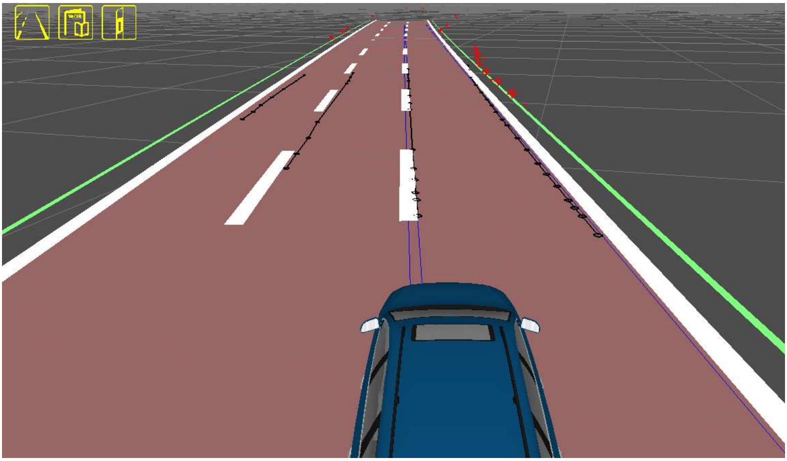


Abbildung 3.7.: Aus Radarreflexen von Leitplanken (rote Punkte) kann eine Schätzung der Randbebauung (grüne Linien) erfolgen. Diese gehen zusammen mit Fahrbahnmarkierungen (schwarze Linienzüge) und Kartendaten (blaue Linien) in die Fahrbahnschätzung (rote Straße) ein.

Insgesamt ist die Fahrbahnfusion für Autobahnen und autobahnähnliche Straßen optimiert, sodass durch ein mehrsegmentiges Klothoidenmodell implizierte Randbedingungen Verwendung finden. Die entsprechenden Klothoidenparameter werden durch ein erweitertes Kalmanfilter geschätzt.

Die Fahrbahnfusion bietet eine Dienstschnittstelle, mit der andere Filter

Informationen über den aktuellen Verlauf der Straße und der relativen Egoposition erhalten. Zusätzlich ermöglicht der Dienst die Transformation von kartesischen Vektoren in das Fahrbahnkoordinatensystem. Dieses ist beispielsweise für Objekte der Objektfusion interessant, welche somit auf der Straße positioniert werden. Für ein Beispiel dazu sei auf Abschnitt 4.6.1 verwiesen.

Belegungsgitterfusion

Die Belegungsgitterfusion betrachtet die Überfahrbarkeit des näheren Umfeldes: Hierzu werden alle generischen Objekt- und Freibereichsdaten beachtet, welche in diese Teilfusion einfließen.

Die Belegungsgitterfusion kann mithilfe einer Polygonschnittstelle abgefragt werden, welche fusionierte Überfahrbarkeitsinformationen für den beschriebenen Bereich liefert. Hierbei ist eine inkrementelle Abfrage möglich, sodass beispielsweise die Befahrbarkeit eines Fahrschlauches in diskreten Schritten mit geringer Latenz analysiert werden kann. Für genauere Informationen zu den verwendeten Fusions- und Abfragealgorithmen sei auf Kapitel 5 verwiesen.

Digitale Karte

Um der Fahrzeugumfeldschätzung eine weitreichende Straßenvorausschau zu ermöglichen, wird eine digitale Karte zusammen mit einer globalen Positions- und Gierwinkelschätzung eingesetzt. Im ersten Schritt wird die Positionsschätzung dazu verwendet, eine Menge von Straßensegmenthypothesen aus der digitalen Karte mithilfe von Abstandsmaßen zu extrahieren. Die Gierwinkelschätzung erlaubt anschließend die Eliminierung unplausibler Hypothesen: Hierzu wird der Gierwinkel des Ego-Fahrzeuges mit dem Richtungswinkel des Straßensegmentes verglichen. Bei Überschreitung eines Schwellwertes wird davon ausgegangen, dass das Ego-Fahrzeug nicht auf dieser Straße fährt: Hierdurch können beispielsweise Straßen, die über Autobahnbrücken führen, von Autobahnen unterschieden werden können.

Die Position und der Gierwinkel werden hauptsächlich durch Satellitennavigationssysteme geschätzt: Entsprechende Empfänger werten dazu in diskreten Zeitabständen Laufzeitmessungen von Satellitensignalen aus und senden die Ergebnisse in die Sensordatenfusion. Um für beliebige Zeitpunkte eine Schätzung zu generieren, werden mithilfe der Egofusion die Positions- und Gierwinkelschätzungen inter- oder extrapoliert. Dadurch können auch kurze Unterbrechungen des GPS-Signals überbrückt

werden, sodass ein Ausfall des Kartendienstes bei temporären Abschattungen vermieden wird. Solche Abschattungen treten beispielsweise bei Tunneldurchfahrten auf.

Insgesamt erlaubt der Kartendienst es anderen Filtern, den Verlauf der aktuellen Straße robust zu ermitteln, wobei die Genauigkeit vom Kartenmaterial und der globalen Positionsschätzung abhängt. Zwar ist in vielen eingesetzten Versuchsträgern diese Schätzung deutlich ungenauer als das Ergebnis der Fahrbahnfusion, jedoch ist die Straßenvorausschau dafür beliebig weit möglich. Zudem können diverse Zusatzinformationen extrahiert werden, wozu neben der Anzahl der Fahrstreifen u. a. auch eine Straßenklasse und die Position von Ampeln, Stoppschildern und Geschwindigkeitsbegrenzungen gehören.

Zusammenfassung

Die vorgestellten Teilfusionen schätzen jeweils unterschiedliche Umfeldaspekte. Hierzu werden einerseits generische Daten aus der Dekodierungsschicht verwendet, andererseits fließen die Ergebnisse anderer Schätzer mit ein. Hierbei ist insbesondere die Eigenbewegungsschätzung zu nennen, die es einerseits ermöglicht, die Dynamik anderer Objekte von der Eigenbewegung zu trennen und andererseits statische Objekte bzw. Fahrbahnmarkierungen aus einem relativen Fahrzeugkoordinatensystem in ein absolutes zu überführen. Die Teilfusionen mit ihren spezifischen Aspekten sind in Tabelle 3.3 aufgeführt.

Teilfusion	Informationsbedarf	Modell
Egofusion	Odometriesensordaten	Einspurmodell
Objektfusion	Egozustand, generische Objektdaten	beschleunigte quaderförmige Körper
Fahrbahnfusion	Egozustand, digitale Karte, generische Objekt- und Fahrbahndaten	mehrsegmentige Klothoiden
Belegungsgitterfusion	Egozustand, generische Objekt- und Freibereichsdaten	Belegungsgitter mit Befahrbarkeitswahrscheinlichkeiten (s. Kapitel 5)
Digitale Karte	Egozustand, GPS-Signal	mit Attributen angereicherte Fahrbahnstützpunkte

Tabelle 3.3.: Teilfusionen der Sensordatenfusion und ihre Spezifika

Die Ego-, Objekt- und Fahrbahnfusion basieren hauptsächlich auf Kalmanfilter, die neben dem Zustand auch dessen Unsicherheit automatisch mitschätzen. Unter den gegebenen Randbedingungen liefern diese Zustandsschätzer bei entsprechender Parametrisierung ausreichend genaue Ergebnisse, was auf eine begrenzte Nichtlinearität der Modelle und auf in etwa normalverteilte Messungen schließen lässt. Diese Teilfusionen haben mit diesen bekannten Ansätzen somit bereits das Potenzial, eine mögliche Genauigkeitsanforderung eines Fahrerassistenzsystems in Echtzeit zu erfüllen, sodass diese im Folgenden nicht genauer untersucht werden.

Zwei andere Schätzer integrieren jedoch komplexe alternative Schätzalgorithmen, die nicht auf Kalmanfilter basieren. Hierbei ist einerseits die Relevanzschätzung für Objekte des Objekttrackers zu nennen und andererseits die Überfahrbarkeitsschätzung mithilfe eines Belegungsgitters. Beide Schätzer werden in den kommenden beiden Kapiteln detailliert vorgestellt.

3.5.6. Enkodierungsschicht: Trennung von Umfeldmodell und Ausgabeprotokoll

Im verwendeten Umfeldmodell existieren sehr viele Zustandsvariablen, die sich hochfrequent ändern. Die kompletten Umfeldmodelldaten zyklisch zur Applikation zu senden ist deshalb sehr aufwendig. Auch würden Änderungen der Schätzrepräsentation Änderungen in der Fahrerassistenzapplikation nach sich ziehen.

Die folgende Enkodierungsschicht trennt daher die Umfeldmodelldarstellung von Fahrerassistenzapplikationen: Hierzu wird zunächst für eine spezifische Applikation ein Protokoll entworfen, welches die benötigten Regelgrößen enthält. Dieses wird von einem sog. „Kollektorfilter“ in der Enkodierungsschicht generiert, wobei dieser die entsprechenden Zustandsvariablen aus dem Umfeldmodell extrahiert und kombiniert. Die Regelgrößen werden mithilfe der Dienstschnittstellen auf den Ausgabezeitpunkt prädiziert, sodass die Abhängigkeit vom genauen Alter des Umfeldmodells minimiert wird.

Als beispielhaftes Fahrerassistenzsystem sei das Hochautomatische Fahren auf Autobahnen genannt, welches eine kontinuierliche Längs- und Querregelung in dieser Domäne ermöglichen soll. Längs- und Querregelung werden im Folgenden als zwei Fahrerassistenzapplikationen aufgefasst. Diese beiden Applikationenwerden von insgesamt vier Protokollen verwendet, welche in Tabelle 3.4 in einer Übersicht dargestellt sind.

So liefert das Egoprotokoll ein Abbild des geschätzten Egozustandes, welches von beiden Applikationen benötigt wird. Zusammen mit dem Quer-

Protokoll	Informationen (Auszug)	Herkunft	Applikation
Ego	Position, Richtung, Geschwindigkeit des eigenen Fahrzeugs	Egofusion	Längs- und Querregelung
Querregelung	Position und Dynamik des Ego-Fahrzeuges im Fahrstreifen	Egofusion, Fahrbahnfusion	Querregelung
Längsregelung	Fahrstreifenbezogene Freibereiche und Kurvenradien vor dem eigenen Fahrzeug	Egofusion, Belegungsgitterfusion, Fahrbahnfusion, digitale Karte	Längsregelung
Objekte	Position und Dynamik anderer Vehrkehrsteilnehmer	Egofusion, Objektfusion, Fahrbahnfusion	Längsregelung

Tabelle 3.4.: Ausgabeprotokolle der Sensordatenfusion

regelungsprotokoll, welches u. a. die Ablage und den Gierwinkel des Ego-Fahrzeuges im Fahrstreifen und eine Fahrbahnkrümmung liefert, ist eine kontinuierliche Querregelung möglich.

Das Längsregelungsprotokoll enthält alle Informationen, die für eine Längsregelung ohne Regelobjekt notwendig ist. Hierzu gehören Geschwindigkeitsbegrenzungen aus der digitalen Karte, eine Vorausschau an zukünftig auftretenden Fahrbahnkrümmungen zur Begrenzung der Querbeschleunigung und für die Anfahrt aus dem Stillstand eine Freiraumüberwachung.

Zusätzlich wird ein Objektprotokoll generiert, welches eine Liste von anderen Verkehrsteilnehmern und statischen Hindernissen repräsentiert. Jedes Objekt wird dabei in den Kontext der Fahrbahn gesetzt und erhält neben einer Dynamikbeschreibung auch eine Relevanzwahrscheinlichkeit (s. Kapitel 4). Diese Objekte können die vom Längsregelungsprotokoll implizierte Höchstgeschwindigkeit weiter beschränken.

Die so generierten Protokolle werden anschließend der Ausgabeschicht übergeben.

3.5.7. Ausgabeschicht: Abstraktion des Sendeweges

Ähnlich wie in der Eingabeschicht kann man die erzeugten Ausgabeprotokolle auf verschiedene Wege aus der Sensordatenfusion senden: Auch in der Ausgabeschicht stehen CAN-Bus, Messdatei und Ethernet für Simulationszwecke zur Verfügung. Für einfache Testzwecke ist es möglich, diese Schicht leer zu lassen, sodass keine Protokollausgabe erfolgt, aber beispielsweise Visualisierungen erzeugt werden.

Zusammen mit der Eingabeschicht lässt sich die Sensordatenfusion in verschiedene Umgebungen transparent einfassen: Neben einer Fahrzeug-Konfiguration und einer Konfiguration zum Abspielen von Messdaten lässt sich die Sensordatenfusion insbesondere auch für „Software-in-the-Loop-Tests (SiL-Tests)“ einbinden. Hierbei werden mithilfe einer virtuellen Fahrt durch eine virtuelle Umgebung Messdaten simulativ erzeugt und per Ethernet zur Sensordatenfusion gesendet. Die Zeitquelle gibt dabei die Simulationsumgebung vor. Die Sensordatenfusion erzeugt die entsprechenden Ausgaben, welche mit der virtuellen Umgebung verglichen werden können.

Insgesamt ermöglichen flexible Ein- und Ausgabeschichten die Modularitätsanforderung zu erfüllen, sodass die Sensordatenfusion verschiedenste Rollen einnehmen kann, ohne dass Unterscheidungen in den dazwischenliegenden Schichten notwendig sind.

3.5.8. Ablaufsteuerung

Viele Assistenzfunktionen benötigen aktuelle Umfelddaten, um eine qualitativ hochwertige Funktion zu gewährleisten: Für die Querregelung beim Hochautomatischen Fahren sind beispielsweise hohe Anforderungen bzgl. Echtzeitfähigkeit zu erfüllen, da der Querregler in kurzen Zeitabständen mit aktuellen Größen des Querregelungsprotokolls versorgt werden muss. Verzögerungen bedeuteten erhöhte Totzeiten im Regelkreis, was direkt eine Verminderung der Regelungsqualität zur Folge hat.

Um die Protokolle rechtzeitig zu generieren, muss die Sensordatenfusion den Kontrollfluss zum Ausgabezeitpunkt an die Kollektorfilter der Enkodierungsschicht übergeben. Hierbei müssen die relevanten Teilfusionen einen konsistenten Zustand aufweisen, damit alle Regelgrößen korrekt extrahiert werden können. Problematisch ist hierbei der externe Einfluss der Umfeldsensoren: Deren Ausgabe kann von der Sensordatenfusion nicht gesteuert werden, sodass eine Teilfusion aufgrund empfangener Sensordaten möglicherweise gerade Schätzalgorithmen ausführt, während aber ein Protokoll mit zeitlich hoher Priorität erzeugt werden soll. Letzteres muss

dann aus Konsistenzgründen auf die Verarbeitung warten, was insgesamt unakzeptable Verzögerungen verursachen kann.

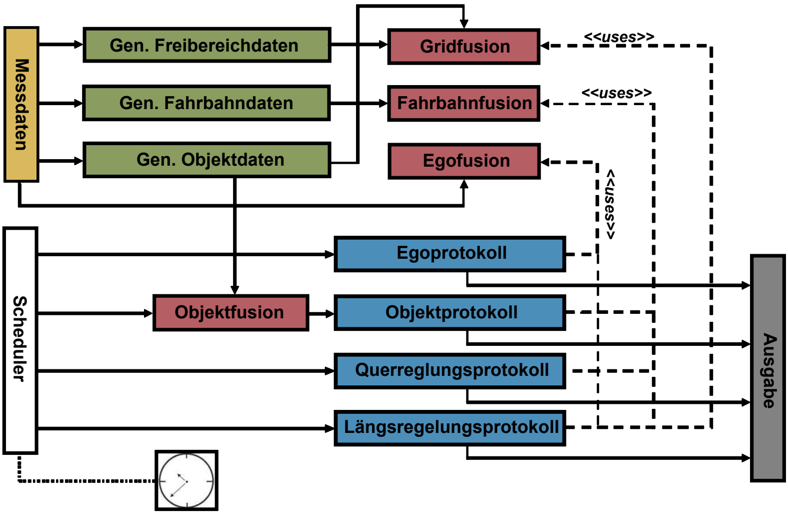


Abbildung 3.8.: Scheduler (weißer Kasten) stößt Protokollfilter (blaue Kästen) bzw. Objektfusion an, sodass zyklische Ausgaben erfolgen

Um den Kontrollfluss unabhängiger von den Sensorausgabezeitpunkten und damit deterministischer zu gestalten, verfügen alle Teilfusionen mit rechenintensiven Schätzalgorithmen Pufferspeicher, in denen sie zunächst alle relevanten Sensordaten sammeln. Diese Teilfusionen erhalten dann den Kontrollfluss, wenn keine Ausgabeprotokolle in nächster Zeit erzeugt werden müssen.

Die Vergabe des Kontrollflusses erfolgt über ein zentrales Filter, welches andere Filter in definierten Zeitschritten anstößt und „Scheduler“ genannt wird. Hierbei wird der aktuelle Zeitpunkt dem angestoßenen Filter übergeben. Eine Übersicht, wie der Scheduler im Filtergraphen der verwendeten Fusionsarchitektur eingebunden ist, ist in Abbildung 3.8 dargestellt.

Der Scheduler vereinfacht einerseits die Erzeugung von zyklischen Ausgaben, andererseits ermöglicht er eine planbare Verteilung der Systemressourcen und definierte Antwortzeiten. Er kann dabei mehrere Ausführungsstränge (engl. „Threads“) verwalten, sodass Multiprozessorplattformen bzw. Mehrkernprozessoren besser ausgenutzt werden können.

Das verwendete Scheduling ist statisch, nicht-präemptiv und prioritätsorientiert: D. h. die Zeitschritte und -abstände der geplanten Prozessaufrufe werden vorher festgelegt und werden zur Laufzeit nicht mehr verändert. Sobald Prozesse gestartet sind, können diese nicht unterbrochen werden, sondern müssen den Kontrollfluss von selbst wieder zurückgeben. Prozesse mit niedriger Priorität (Visualisierung, Fusionierung) erhalten keinen Kontrollfluss, wenn Prozesse mit hoher Priorität (Erzeugung von Ausgabeprotokolle) anstehen.

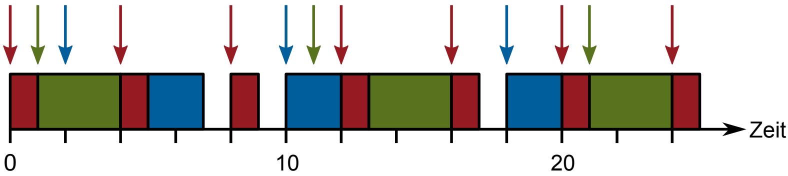


Abbildung 3.9.: Nicht-präemptives statisches Scheduling für drei Prozesse mit unterschiedlicher Priorität. Pfeile symbolisieren den jeweils geplanten Start, Kästen die zeitliche Prozessorauslastung (ein Kern).

Ein Beispiel für solch einen Schedule mit drei Prozessen (rot/grün/blau) ist in Abbildung 3.9 dargestellt. Diese weisen unterschiedliche geplante Startzeiten (0/1/2), Zykluszeiten (4/10/8), Rechendauer (1/3/2) und Prioritäten (hoch/niedrig/niedrig) auf. In dem Beispiel werden die Prozesse häufig zeitverzögert gestartet, was u. a. mit einer hohen durchschnittlichen Prozessorkernauslastung von 80% begründet ist.

Auf der Zielplattform liegt die Prozessorkernauslastung trotz einer deutlich höheren Anzahl an Prozessen deutlich niedriger[10], sodass solche Verzögerungen durch ein geschickt gewähltes statisches Schedule größtenteils vermieden werden können. Je mehr Prozesse und je höher die durchschnittliche Systembelastung, desto schwieriger ist es allerdings, ein gutes oder gar optimales Schedule zu finden: So ist nach [Garey 1979] das Entscheidungsproblem, ob ein optimales Schedules für nicht unterbrechbare Prozesse existiert, im Allgemeinen NP-vollständig. Durch die Verwendung eines Mehrkernprozessors und durch die geringe Anzahl von Prozessen mit hoher Priorität konnte jedoch problemlos ein Schedule mit ausreichender Qualität für diese Sensordatenfusionsinstanz erstellt werden.

[10] Die vorgestellte Sensordatenfusion verwendet 19 Prozesse, welche den Prozessor auf der Zielplattform im Schnitt zu 17,5% auslasten

Der Scheduler bildet eine wesentliche Komponente zur Erfüllung der Echtzeitanforderung. Er ermöglicht eine kontrollierte Rechenlastverteilung und die zyklische Protokollausgabe zu definierten Zeitpunkten. Durch eine interne Statistikkomponente erleichtert er zudem die Bewertung der Echtzeitfähigkeit verwendeter Algorithmen, indem er die Zeit misst, bis ein Filter den Kontrollfluss wieder zum Scheduler zurückgibt.

3.6. Anwendungen und erreichte Modularität

Die Modularität und lose Kopplung der Sensordatenfusionarchitektur erlaubt die einfache Adaption an neue Sensoren und Funktionen und die Portierung auf verschiedene Versuchsträger. Von dieser Eigenschaft wurde intensiv Gebrauch gemacht, indem die Sensordatenfusion bisher auf acht verschiedenen Versuchsfahrzeugen portiert worden ist. Dort bedient die Sensordatenfusion unterschiedlichste Fahrerassistenzfunktionen[11]:

- Hochautomatisches Fahren auf der Autobahn und autobahnähnlichen Straßen [Steinmeyer 2010a; Hoeger 2011]
- Vorausschauende energieeffiziente Fahrzeuglängsregelung[12] [Dornieden 2012]
- Nothalteassistenz bei medizinisch bedingter Fahrunfähigkeit [Mirwaldt 2012]
- Fahrstreifenwechselassistenz [Degerman 2012; Meinecke 2013], Fahrstreifenhalteassistenz und hochautomatisches Fahren in Stausituationen für schwere Nutzfahrzeuge

Die Versuchsträger weisen je nach Anforderungen der Assistenzfunktionen unterschiedlichste Sensorkonfigurationen auf: So finden sich alle im Kapitel 2 erwähnten Sensortechnologien in diesen wieder. Für die Egoschätzung kommen in einigen Versuchsträgern Inertialplattformen zum Einsatz, während bei anderen Fahrzeugen ausschließlich die serienmäßigen Odometriesensoren genutzt werden.

[11] Aufgrund der vielen verschiedenen Funktionen, Sensoren und Versuchsträger wird in dieser Arbeit auf die Darstellung von spezifischen Anforderungen und Gesamtarchitekturen verzichtet. Es sei aber darauf verwiesen, dass diese für eine hochautomatische Fahrfunktion mit einer Sensorkonfiguration im Rahmen des EU-Förderprojekt HAVEit dokumentiert worden sind [To 2010b,a; Hoeger 2011].

[12] Hier kam eine vereinfachte Version der Sensordatenfusion zum Einsatz, da es nur geringe Anforderungen an die Genauigkeit der Fahrstreifenschätzung gab und auch kein Belegungsgitter notwendig war.



(a) Versuchsträger für das HAVEit-Projekt ausgerüstet mit Laserscanner, Radarsensor, Ultraschallsensoren, GPS und digitaler Karte



(b) Versuchsträger ausgerüstet mit Stereokamera, zwei Monokameras, fünf Radarsensoren, zehn Ultraschallsensoren, zwei Inertialplattformen, GPS und digitaler Karte

Abbildung 3.10.: Zwei Versuchsträger mit unterschiedlichen Sensorkonfigurationen (Bild aus [Meinecke 2013])

Zwei Beispiele für verwendete Versuchsträger sind in Abbildung 3.10 dokumentiert: Diese unterstützen hochautomatische Fahrfunktionen bzw. Fahrstreifenwechsel- und Fahrstreifenhalteassistenz. Obwohl in beiden Versuchsträgern viele neue Sensoren verbaut worden sind, sind die Verbauorte nicht direkt offensichtlich. Der Hintergrund dafür ist, dass nur solche Verbauorte ausgewählt worden sind, die auch für einen späterem Serieneinsatz in Frage kämen. Deshalb fällt beispielsweise das Autodach als Sensorverbauort aus, obwohl Sensoren dort sehr große Sichtbereiche abdecken könnten.

3.7. Zusammenfassung

Die gezeigte Softwarearchitektur tariert die Komplexität einer generischen Sensordatenfusion mit der benötigten Flexibilität aus. Wesentliche Merkmale sind die lose und konfigurierbare Kopplung von Filtern und die Zusammenfassung in Schichten mit genau definierten Verantwortlichkeiten.

So ist es mit möglich, die Sensordatenfusion auf andere Versuchsträger mit unterschiedlicher Sensorik zu portieren. Hier ist im Wesentlichen die Sensorabstraktionsschicht anzupassen, welche die sensorspezifischen Protokolle auf generische Strukturen abbildet.

Ist eine Erweiterung des Umfeldmodells mit neuen Schätzern notwendig, so sind diese in die Fusionsschicht hinzuzufügen. Da diese als Dienste für viele andere Filter fungieren, ist ein Austausch oder eine Simulation vorhandener Schätzer durch eine entsprechende Reimplementierung der Dienstschnittstelle effizient möglich. Durch Rekonfiguration des Filtergraphens lassen sich flexibel die gewünschten Fusionsteile neu zusammenstellen.

Die Ausgabeprotokolle können leicht und flexibel an neue Fahrerassistenzapplikationen angepasst werden, indem die Filter in der Enkodierungsschicht die gewünschten Attribute aus dem Umfeldmodell extrahieren und herausschreiben. Die gesamte Fusion kann in beliebigen Umgebungen laufen; hierzu sind Eingabe- und Ausgabeschicht entsprechend anzupassen, wovon zu Simulationszwecken intensiv Gebrauch gemacht wird.

Die Filterarchitektur unterstützt dabei die Softwareentwicklung im Team, indem Filter unabhängig von anderen Filtern entwickelt werden können. So ist die parallele Entwicklung verschiedener Filterkomponenten möglich, ohne dass unterschiedliche Versionen von Sensordatenfusionsoftware zusammengeführt werden müssen.

Anforderung	**Exemplarische Umsetzung**
Genau	Detaillierte Modelle (vgl. Kapitel 5), Kalmanfilter
Robust	Ablaufsteuerung, Zustandsprädiktion, Störungsneutralisation mittels erlernter Modelle (vgl. Kapitel 4)
Probabilistisch	Kalmanfilter, Relevanzschätzer (vgl. Kapitel 4), probabilistische Belegungsgitterfusion (vgl. Kapitel 5)
Echtzeitfähig	Ablaufsteuerung, geringe algorithmische Komplexität, Retrodiktions- und Prädiktionsfähigkeit
Modular	Generische Daten, Schichtenmodell, Pipe-and-Filters-Architektur, Teilschätzer als Dienste

Tabelle 3.5.: Anforderungen an eine Sensordatenfusion und exemplarische Umsetzungen

Des Weiteren ermöglicht das „Pipe-and-Filters“-Architekturmuster die Parallelisierung der Sensordatenfusion, sodass diese eine Multiprozessorplattform bzw. Mehrkernprozessoren besser ausnutzen kann. Durch

das explizite Scheduling sind Echtzeitanforderungen in einzelnen Verarbeitungssträngen kontrolliert umsetzbar. So lassen sich verschiedene Filter zu definierten Zeiten auf solchen Plattformen parallel anstoßen, sodass die verfügbare Rechenkapazität optimal zur Latenzzeitoptimierung ausgenutzt werden kann. Dieses ist beispielsweise für die Güte einer Querregelung von erheblicher Bedeutung. Eine Übersicht, wie die allgemeinen Anforderungen in der Sensordatenfusion berücksichtigt werden, ist in Tabelle 3.5 dokumentiert.

Die erfolgreiche Umsetzung der Anforderungen zeigt sich dadurch, dass die entworfene Sensordatenfusion nicht nur auf mittlerweile acht Versuchsträger mit unterschiedlicher Sensorik portiert werden konnte, sondern dass diese auch verschiedenste Fahrerassistenzapplikationen bewältigen.

Insgesamt hat sich dieses Kapitel hauptsächlich mit der Modularitäts- und Echtzeitsfähigkeitsanforderung befasst, wobei auch exemplarisch einige Teilfusionen vorgestellt wurden, die größtenteils auf Kalmanfilter basieren. Diese zeichnen sich durch eine gute Zustandsschätzung und Unsicherheitsberechnung aus und bilden damit bereits eine wichtige Säule zur Erfüllung der Anforderung an eine genaue, robuste und probabilistische Sensordatenfusion. Der Kalmanfilter ist jedoch kein idealer Schätzer für alle möglichen Umfeldaspekte. Deshalb werden weitere Methoden zur der Umfeldschätzung in den nächsten beiden Kapiteln vorgestellt.

Kapitel 4.

Existenz- und Signifikanzschätzung für Umfeldobjekte

4.1. Einleitung

Mit der im vorherigen Kapitel vorgestellten Objektfusion ist es möglich, Objektdaten von verschiedenen Sensoren durch Kalmanfilter zu einer Objektliste zu fusionieren. Durch diese Filterung wird die Zustandsunsicherheit verkleinert und quantifiziert. Insgesamt ist durch diese Methode das Problem der Messungenauigkeiten von Sensoren behandelt worden.

Die resultierende Objektliste beinhaltet jedoch weitere Unsicherheiten: Wie in Kapitel 2 erwähnt worden ist, können vermeintliche Objekte erfasst werden, welche jedoch keine relevanten Objekte im Sinne einer Fahrerassistenzapplikation darstellen, da sie beispielsweise über- oder unterfahrbar sind. Umgekehrt ist es möglich, dass unter Umständen wichtige Objekte nicht oder nicht in allen Messungen detektiert wurden. In Kapitel 3 sind weitere Fehlerquellen bei der Objektverfolgung erwähnt worden. So können durch Assoziationsfehler sowohl Scheinobjekte entstehen als auch Objekte unerkannt bleiben.

Insgesamt bleibt bei widersprüchlichen und ungenauen Sensordaten die Bewertung schwierig, ob eine Objekthypothese von der Fahrerassistenzapplikation berücksichtigt werden muss oder nicht. Um diesem zu begegnen, befasst sich dieses Kapitel mit der statistisch genauen Bewertung der Objektdaten der vorgestellten Sensordatenfusion. Für jede Objekthypothese wird eine Wahrscheinlichkeit bestimmt, welche angibt, ob dieses im Sinne des Fahrerassistenzsystems berücksichtigt werden muss.

Die probabilistische Ausgabe ist nicht nur Ausdruck und Quantifizierung der Imperfektion der Umfeldwahrnehmung. Ihr Nutzen besteht in der Anwendungsmöglichkeit von Schwellwerten, sodass beispielsweise die Fehlauslösungsrate auf Kosten der fehlerhaften Nichtauslösungsrate vorhersagbar gesenkt werden kann oder umgekehrt.

Als Domäne werden in diesem Kapitel Autobahnen und autobahnähnliche Szenarien gewählt, da diese besonders strukturiert und somit für

Ansätze mit vielen Modellannahmen geeignet sind. Diese werden in diesem Kapitel untersucht. Für schwach strukturierte Umgebungen eignen sich belegungsgitterbasierte Ansätze, welche detailliert im Kapitel 5 vorgestellt werden. Herausfordernd bei der angenommenen Autobahndomäne ist insbesondere die sichere Objektdetektion in großen Entfernungen. Die nachfolgend vorgestellten Methoden können aber auch auf andere Domänen übertragen werden.

4.2. Gliederung

In diesem Kapitel werden zunächst verschiedene Typen von Objekthypothesen definiert, welche existente bzw. signifikante Objekte repräsentieren können. Sind diese Objekte sowohl existent als auch signifikant, so müssen diese vom Fahrerassistenzsystem berücksichtigt werden und gelten als relevant. Genaue Definitionen zur Existenz, Signifikanz und Relevanz erfolgen im nächsten Abschnitt 4.3.

Die Typklassifikation der Objekthypothesen erfolgt durch Schätzer, welche ausschließlich auf mehr oder weniger unsichere und ungenaue Informationen zurückgreifen und somit keine perfekten Ergebnisse liefern können. In den folgenden Abschnitten werden bekannte und neue Ansätze vorgestellt, die eine solche Schätzung leisten können.

Zunächst werden bekannte Ansätze aus der Literatur untersucht, wobei insbesondere zur Existenzschätzung eine Vielzahl von Ansätzen entwickelt worden sind. Einige Vielversprechende davon werden im Abschnitt 4.4 vorgestellt und bewertet. Hier stellt sich heraus, dass viele Ansätze Objektexistenz als dynamischen Zustand betrachten und deshalb entsprechende Schätzer nutzen. Nur wenige Ansätze sehen Existenzschätzung als auf Klassifikationsproblem und diese arbeiten nicht probabilistisch.

Weiterhin werden verschiedene Signifikanzschätzer speziell für Abstandsregeltempomaten (ACC) im Abschnitt 4.5 vorgestellt und untersucht.

Existenz- und Signifikanzschätzungen müssen kombiniert werden, um zu entscheiden, ob eine Objekthypothese relevant ist. Als alternativer Ansatz wird ein integrierter Relevanzschätzer vorgestellt, welcher nicht über den Umweg von Existenz- und Signifikanzschätzung arbeitet, sondern die Relevanz direkt ermittelt. Dieser approximiert mittels Maschinellem Lernen eine Klassifikatorfunktion aus Trainingsdaten, womit er alle vorgestellten und auch unbekannten Fehlerquellen integriert. Dieser Ansatz wird im Abschnitt 4.6 vorgestellt, an realen Messdaten getestet und bewertet.

Die Transparenz von Klassifikatoren, die mittels maschinellen Lernverfahrens erzeugt werden, ist häufig gering. Um zu analysieren, welche Zu-

sammenhänge dieser Klassifikator gelernt hat, werden im Abschnitt 4.7 Schnittbilder durch den Zustandsraum eines solchen Klassifikators berechnet und interpretiert.

Die dargestellten Ansätze, Ergebnisse und Bewertungen dieses Kapitels werden im letzten Abschnitt 4.8 zusammengefasst.

4.3. Existente und signifikante Objekte

Die im Kapitel 3 erzeugte Objektliste besteht aus einer Menge von Objekthypothesen. Jede enthält einen Zustandsvektor, der u. a. Dynamik und Position des Objektes dokumentiert. Es fehlt allerdings eine Schätzung, ob sich hinter dem Objekt ein Scheinobjekt oder ein Ziel im Sinne des Fahrerassistenzsystems verbirgt.

Deshalb ist die Bestimmung der Objektexistenz für jede Objekthypothese der Sensordatenfusion notwendig. Diese unterscheidet die Klasse der Scheinziele (z. B. Schilderbrücke) von der Klasse der existenten Ziele (z. B. Fahrzeug) ausdrücklich im Sinne des Fahrerassistenzsystems. Dieses Vorgehen betont, dass nicht die Fahrerassistenzapplikation an eine Umfeldwahrnehmung angepasst werden soll, sondern dass die Fahrerassistenzapplikation Anforderungen an die Umfeldwahrnehmung stellt, die es zu erfüllen gilt[1]:

Definition 1 (Objektexistenz)**.**
Eine Objekthypothese wird im Sinne einer Fahrerassistenzapplikation genau dann als existent klassifiziert, wenn diese zu einem realen Objekt korrespondiert, welches unabhängig vom aktuellen Zustand einen Typ aufweist, auf den das Fahrerassistenzsystem potenziell reagieren muss.

Ist das Objekt existent, so bedeutet das aber noch nicht, dass die Fahrerassistenzapplikation direkt auf dieses reagieren muss. Zusätzlich muss noch geprüft werden, ob dieses auch signifikant im Sinne des Fahrerassistenzsystems ist. So sind beispielsweise Fahrzeuge für ein ACC nicht signifikant, wenn sich diese außerhalb des Fahrschlauches befinden. Im Bereich Fußgängerschutz sind nur die Fußgänger signifikant, welche sich im unmittelbaren Nahbereich aufhalten.

Definition 2 (Objektsignifikanz)**.**
Eine Objekthypothese wird im Sinne einer Fahrerassistenzapplikation genau dann als signifikant klassifiziert, wenn das zugrundeliegende Objekt

[1] In [Munz 2011, S. 31 f] wird eine ähnliche Definition zur Existenz von Objekten aufgestellt. Munz fordert allerdings zusätzlich die Detektierbarkeit durch mindestens einen Sensor des Egofahrzeuges. Darauf wird hier bewußt verzichtet, da die Objektexistenz hier unabhängig von der Beobachtbarkeit sein soll.

einen Zustand aufweist, bei dem das Fahrerassistenzsystem bei angenommener Existenz reagieren muss.

Offensichtlich sind für Fahrerassistenzsysteme nur solche Objekte zu berücksichtigen, welche existent und signifikant sind. Diese beiden Eigenschaften werden zur Objektrelevanz zusammengefasst:

Definition 3 (Objektrelevanz)**.**
Eine Objekthypothese ist im Sinne einer Fahrerassistenzfunktion genau dann relevant, wenn diese existent und signifikant ist.

Da die Information einzelner Sensoren sowohl zur Objektexistenz als auch Objektsignifikanz aus potenziell unsicheren Sensordaten stammen, ist eine jederzeit fehlerfreie Klassifikation praktisch nicht möglich, da zwischen Objekthypothese und zugrunde liegendem Objekt stets eine mehr oder weniger große Abweichung besteht. Aus diesem Grund sollten möglichst viele Informationsquellen genutzt werden, um die Unsicherheiten bei der Schätzung von Existenz, Signifikanz und Relevanz eines Objektes zu minimieren. Die Stärke der Unsicherheit sollte durch ein Wahrscheinlichkeitsmaß möglichst genau wiedergegeben werden.

Dieser probabilistische Ansatz ermöglicht die Auflösung und Quantifizierung von widersprüchlichen Informationen und liefert zur Klassifikation zusätzlich eine Schätzgüte.

4.4. Stand der Technik: Existenzwahrscheinlichkeitsschätzer

Existenzwahrscheinlichkeitsschätzer haben im Wesentlichen zwei Aufgaben. Zum einen müssen sie Objekte vom Hintergrundrauschen separieren, sodass Scheinobjekte ohne kausalen Zusammenhang in der Umwelt unterdrückt werden.

Wie in Kapitel 2 gezeigt, gibt es zum anderen Phänomene in der Umwelt, welche keine Objekte im Sinne des Fahrerassistenzsystems sind, aber dennoch systematische Quellen für eine positive Objekthypothese darstellen können. Diese zu eliminieren ist die zweite und wohl anspruchsvollere Aufgabe eines Existenzwahrscheinlichkeitsschätzers.

In diesem Abschnitt werden insgesamt sechs verschiedene existierende Ansätze vorgestellt und bewertet, wobei fünf bekannte Ansätze aus der Literatur sind.

4.4.1. Track Score Funktion nach Sittler

Eine der ersten Arbeiten zur Bewertung von Objekthypothesen mittels eines Gütemaßes ist in [Sittler 1964] zu finden, welches später in [Blackman 1999] und [Altendorfer 2010] weiter vertieft wird. Ursprünglich wurde dieses Maß als Entscheidungshilfe im Tracking verwendet, um Messungen mit Objekthypothesen besser assoziieren zu können. Es kann außerdem dazu verwendet werden, den Lebenszyklus einer Objekthypothese zu verwalten: Sinkt das Gütemaß unter einen Schwellwert, so wird die Objekthypothese verworfen.

Die vorgeschlagene Gütefunktion („Likelihood Ratio") gibt den Quotienten der Wahrscheinlichkeiten $P(H_1|M^k)$ und $P(H_0|M^k)$ an, dass ein Objekt existiert bzw. nicht existiert. Hierzu wird eine Wahrscheinlichkeitsdichtefunktion erzeugt, welche die Existenzwahrscheinlichkeit aufgrund eines Messvektors und der im Tracking erzeugten zugehörigen Kovarianzmatrix abbildet. Über einen Markov-Ketten-Ansatz wird die Wahrscheinlichkeit für die beiden Hypothesen bestimmt, ob das Objekt existiert bzw. nicht existiert.

Sei $M^k = \{m_1, \ldots, m_k\}$ die Menge der Messungen, H_1 bzw. H_0 die Hypothese, dass das Objekt existiert bzw. nicht existiert und $P_0(H_1)$ bzw. $P_0(H_0)$ die Grundwahrscheinlichkeit eines Tracks, dass es sich um ein existierendes Objekt bzw. um ein Scheinobjekt handelt.

Dann sei eine „Likelihood Ratio"-Funktion LR definiert, die das Verhältnis der Wahrscheinlichkeiten P_{H_1} und P_{H_0} beschreibt, ob ein Track existiert bzw. nicht existiert:

$$LR(M^k) = \frac{p(M^k|H_1) \cdot P_0(H_1)}{p(M^k|H_0) \cdot P_0(H_0)} = \frac{p(M^k|H_1) \cdot P_0(H_1)}{1 - p(M^k|H_1) \cdot P_0(H_1)} \stackrel{\Delta}{=} \frac{P(H_1|M^k)}{P(H_0|M^k)} \tag{4.1}$$

Diese Funktion basiert auf einer Unabhängigkeitsannahme der einzelnen Messungen bezüglich der beiden Objekthypothesen H_1 und H_0:

$$p(M^k|H_1) = \prod_{i=1}^{k} p(m_i|H_1) \tag{4.2}$$

Herausfordernd bleibt die genaue Modellierung der Wahrscheinlichkeitsdichtefunktion basierend auf einem potenziell hochdimensionalen Messvektor. Hierzu wird in [Blackman 1999] die LR-Funktion in die Teilfunktionen LR_K und LR_S aufgetrennt, die die unterschiedlichen Existenzwahrscheinlichkeiten bzgl. Kinematik des beobachteten Tracks und für die Signalcharakteristik beschreiben. Dieses Vorgehen vereinfacht die Modellierung der korrespondierenden Wahrscheinlichkeitsdichtefunktion.

In die auf kinematischen Attributen basierende Wahrscheinlichkeitsdichtefunktion fließen verschiedene Schätzungen aus dem Tracking und der assoziierten Messung ein: Hierzu gehört die Determinante der Kovarianzmatrix S des Zustandsvektors z (s. Abschnitt A.2). Je geringer diese ist, desto geringer ist die Zustandsunsicherheit und desto häufiger wurde das Objekt mit übereinstimmenden Messungen bestätigt. Des Weiteren wird der mit der Kovarianzmatrix gewichtete Mahalanobisabstand $d(m_i,S)^2$ der assoziierten Messung $\tilde{m}_i = Hm_i$ zum Trackzustand mit $d(m_i,S)^2 = \tilde{m}_i\, S^{-1}\, \tilde{m}_i$ berücksichtigt. Als letztes Attribut fließt ein messwertabhängiger Korrekturfaktor $V_C(m_i)$ ein, der für einen diskreten Bereich im Zustandsraum die unterschiedliche Auftretenswahrscheinlichkeit von Scheinzielen kompensiert.

Die Teilfunktion LR_K ergibt sich zu:

$$LR_K(m_i,S) = \frac{V_C(m_i)\cdot e^{\frac{-d(m_i,S)^2}{2}}}{(2\pi)^{\frac{\dim(m_i)}{2}}\sqrt{|S|}} \tag{4.3}$$

Die Modellierung der Wahrscheinlichkeitsdichtefunktion für die Berechnung von LR_S ist abhängig von der Sensortechnologie. In [Blackman 1999] wird beispielsweise das ermittelte Signal-Rausch-Verhältnis eines Radarsensors während der Signalverarbeitung genutzt, um diese zu ermitteln.

Allgemein wird ein Signalvektor $m_S(m)$ aus dem Messvektor m generiert, der die signalspezifischen Daten enthält; $m_A(m)$ ist die aus dem Messvektor generierte Amplitude und TH eine Schwelle, ab der m_A als positive Detektion klassifiziert wird. Dann ergibt sich LR_S zu:

$$LR_S(z) = \frac{p(m_S(z), m_A(z) \geq TH|H_1)}{p(m_S(z), m_A(z) \geq TH|H_0)} \tag{4.4}$$

Die Zusammenfassung von LR_K und LR_S eines Tracks, mit dem k Einzelmessungen assoziiert worden sind, erfolgt durch:

$$LR(M^k,S^k) = \frac{P_0(H_1)}{P_0(H_0)} \prod_{i=1}^{k} (LR_K(m_i,S_i)\cdot LR_S(m_i)) \tag{4.5}$$

Um aus dem LR-Funktionsergebnis eine Existenzwahrscheinlichkeit zu bestimmen, kann man durch Umformung der Gleichung 4.1 und der totalen Wahrscheinlichkeit mit $P(H_1|M^k) + P(H_0|M^k) = 1$ folgende Beziehung ableiten:

$$P(H_1|M^k,S^k) = \frac{LR(M^k,S^k)}{LR(M^k,S^k)+1} \tag{4.6}$$

In der Literatur wird häufig die „Logarithmic Likelihood Ratio“(LLR) angegeben, welche durch Logarithmieren der LR-Funktion erzeugt wird. Diese erlaubt einen summarischen Aktualisierungsalgorithmus. $LLR(M^k, S^k)$ und $P(H_1|M^k)$ ergeben sich dann zu:

$$LLR(M^k, S^k) = \ln(LR(M^k, S^k)) \tag{4.7}$$

$$P(H_1|M^k, S^k) = \frac{e^{LLR(M^k,S^k)}}{e^{LLR(M^k,S^k)} + 1} \tag{4.8}$$

Die von Sittler vorgeschlagene Gütefunktion berücksichtigt bereits viele Faktoren, insbesondere alle Attribute des Messvektors m und die Zustandsunsicherheit innerhalb des Trackings in Form der Determinante der Kovarianzmatrix. Somit lassen sich viele wesentliche Attribute in diesem Gütemaß vereinen, sodass dieses potenziell eine hohe Funktionsapproximation erreichen kann.

Nachteilig an diesem Verfahren ist die Erfordernis der genauen Modellierung der beiden Wahrscheinlichkeitsdichtefunktionen LR_K und LR_S. Während bei LR_K nur die Scheinzielkompensation V_C aufgrund eines potenziell hochdimensionalen Messvektors aufwendig zu ermitteln erscheint, ist bei LR_S sehr viel sensorspezifisches Expertenwissen notwendig, um eine ausreichende Approximation der Wahrscheinlichkeitsdichtefunktion zu erreichen.

Eine wohl größere Fehlerquelle liefert jedoch die Unabhängigkeitsannahme zwischen einzelnen Messungen entsprechend Gleichung 4.2. Dadurch können systematische Scheinziele, wie beispielsweise eine Schilderbrücke, in diesem Verfahren nicht berücksichtigt werden. Sie werden nur unzureichend über die Scheinzielrate V_C gemittelt kompensiert. So wird für dieses Verfahren nach [Blackman 1999] auch eine gleichverteilte Auftretenswahrscheinlichkeit für Scheinziele in einem Teilbereich im Zustandsraum V_C und für die gemessenen Objektzustände eine Normalverteilung vorausgesetzt.

4.4.2. Trackqualität nach Sinha

Ein weiteres Gütemaß zur Schätzung der „Trackqualität“ ist in [Sinha 2006] angegeben. Die Schätzung soll ebenfalls mit der Existenzwahrscheinlichkeit des Tracks korrelieren.

Wird ein Track das erste Mal initialisiert, erhält er eine konstante oder messwertabhängige Startexistenzwahrscheinlichkeit P_0. Die Prädiktion folgt einem einfachen Markov-Modell gemäß Abbildung 4.1, welches

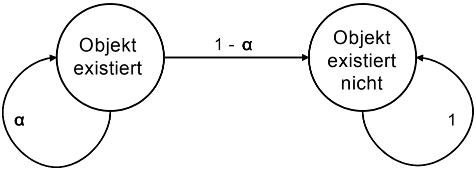


Abbildung 4.1.: Markov-Kette nach Sinha für Ereignis A_1

durch einen einzelnen konstanten Faktor α parametrisiert ist. Dieser beschreibt die Wahrscheinlichkeit, dass ein existierendes Objekt im nächsten Prädikationsschritt noch existiert.

Gemäß dem Markov-Modell kann sich eine Objekthypothese mit der Wahrscheinlichkeit von $1-\alpha$ von existent zu nicht existent wandeln, jedoch nicht umgekehrt. Sinha nimmt an, dass eine Objekthypothese im Messrauschen nach einer gewissen Trackingdauer als existierendes Objekt bestätigt werden kann und dann nur eine gewisse Zeitspanne existiert, bevor es einen Ort einnimmt, wo es nicht mehr detektierbar ist. Die Nichtdetektierbarkeit setzt Sinha mit der Nichtexistenz gleich.

Insgesamt ergibt sich durch das Markov-Modell eine einfache Prädiktion pro Zeitschritt. Gegeben sei ein Track mit der Existenzwahrscheinlichkeit $P(k|k)$ und ein Persistenzfaktor α, welcher die Wahrscheinlichkeit beschreibt, dass ein existenter Track immer noch existiert, wenn dieser im aktuellen Messzyklus nicht bestätigt werden konnte. Die prädizierte Existenzwahrscheinlichkeit beträgt dann:

$$P(k|k+1) = P(k|k)\alpha \tag{4.9}$$

Im Aktualisierungsschritt werden die beiden Ereignisse A_1 und A_2 unterschieden: Entweder konnte der Track mit einer Messung assoziiert werden oder nicht. Diese beiden Ereignisse A_1 und A_2 lassen sich noch in 5 unbekannte Teilereignisse $A_1 = \{A_{11}, A_{12}, A_{13}\}$ und $A_2 = \{A_{21}, A_{22}\}$ unterteilen, welche zwei Attribute mit unbekanntem Zustand enthalten.

Das eine Attribut besagt, ob das getrackte Objekt tatsächlich existiert. Das andere Attribut beinhaltet, ob der Sensor ein existierendes Objekt wahrgenommen hat oder aber ein falsch positives Objekt durch eine fehlinterpretierte Messung geliefert hat. Die Zusammensetzung der 5 Teilereignisse ist in Tabelle 4.1 dokumentiert.

Sinha nimmt an, dass die Ereignisse A_{11}, A_{12}, A_{13}, in denen eine Messung erfolgreich assoziiert werden konnte, disjunkt und vollständig sind.

	A_{11}	A_{12}	A_{13}	A_{21}	A_{22}
Assoziation möglich	Ja	Ja	Ja	Nein	Nein
Objekt existiert	Ja	Ja	Nein	Ja	Nein
Messung stammt von einem existierenden Objekt	Ja	Nein	Nein	Nein	Nein

Tabelle 4.1.: Auftretende Teilereignisse bei einer Messassoziation

Der Fall, dass eine auf einem existierenden Objekt basierende Messung eine Fehlassoziation verursacht, wird vernachlässigt.

Auf Basis der Auftretenswahrscheinlichkeit dieser Teilereignisse wird die neue Existenzwahrscheinlichkeit $P(k+1|k+1)$ bestimmt. Bei der Berechnung wird eine Fallunterscheidung vorgenommen, ob das Ereignis A_1 (Objekt assoziiert) oder das Gegenereignis A_2 eingetreten ist:

$$P(k+1|k+1) = \begin{cases} \frac{p_{A_{11}}+p_{A_{12}}}{p_{A_{11}}+p_{A_{12}}+p_{A_{13}}} & \text{, wenn } A_1 \text{ eingetreten ist} \\ \frac{p_{A_{21}}}{p_{A_{21}}+p_{A_{22}}} & \text{, sonst} \end{cases}$$

Hierbei sind die Wahrscheinlichkeiten $p_{A_{11}}$, $p_{A_{12}}$, $p_{A_{13}}$, $p_{A_{21}}$ und $p_{A_{22}}$ der einzelnen Teilereignisse genau zu modellieren. Um eine gute Funktionsapproximation zu erreichen, sollten möglichst viele relevante Eingangsparameter genutzt werden.

Dafür verwendet Sinha die a-priori zu bestimmenden Detektionswahrscheinlichkeiten π_1 und π_2 für die Ereignisse A_1 bzw. A_2 und die Wahrscheinlichkeit π_0, dass kein Scheinziel im Bereich einer Objekthypothese aufgetreten ist. Als weitere Funktion wird $f(z_{k+1}|x_{k+1})$ definiert, welche die Wahrscheinlichkeit bestimmt, dass die Messung z_{k+1} zu der auf den Zeitpunkt der Messung prädizierten Objekthypothese x_{k+1} gehört. Wie die im vorherigen Abschnitt vorgestellte „Track Score Function" von Sittler, wird als weiterer Faktor noch die Auftretenswahrscheinlichkeit von Scheinzielen $V_C(z)$ abhängig vom Messvektor z berücksichtigt:

$$p_{A_{11}} = \pi_1 \cdot f(z_{k+1}|x_{k+1}) \cdot P(k+1|k) \quad (4.10)$$

$$p_{A_{12}} = (1-\pi_1) \cdot V_C(z_{k+1}) \cdot P(k+1|k) \quad (4.11)$$

$$p_{A_{13}} = V_C(z_{k+1}) \cdot (1-P(k+1|k)) \quad (4.12)$$

$$p_{A_{21}} = (1-\pi_2) \cdot \pi_0 \cdot P(k+1|k) \quad (4.13)$$

$$p_{A_{22}} = \pi_0 \cdot (1-P(k+1|k)) \quad (4.14)$$

Der vorgestellte Algorithmus zur Berechnung der Trackqualität von Sinha kann ähnliche Faktoren wie die „Track Score Function“ von Sittler nutzen. Die Funktion f zur Berechnung der Wahrscheinlichkeit, ob eine Messung einem Track angehört, ist in [Sinha 2006] nicht weiter spezifiziert. Ein offensichtlicher Ansatz ist den mit der Kovarianzmatrix gewichteten Mahalanobisabstand von Messvektor und prädizierter Objekthypothese zu nutzen. Auch die Verwendung sensorspezifischer Gütefunktionen, wie die im vorherigen Abschnitt vorgestellte LR_S-Funktion, ist denkbar. Weiterhin gemeinsam haben beide Funktionen, dass sie die gemittelte Auftretensrate von Scheinzielen in definierten Teilbereichen des Zustandsraumes verwenden.

Nachteilig ist die eventuell komplexe oder ungenügend approximative Modellierung von f. Wie bei Sittler ist auch die messvektorabhängige Scheinzielrate V_C bei hochdimensionalen Messvektoren eventuell nur aufwendig zu ermitteln. Auch das Problem der systematischen Scheinziele oder systematisch nicht erkannten Ziele wird nur unzureichend berücksichtigt. Diese werden hier, wie die gemittelte Scheinzielrate zeigt, als gleichverteilt angenommen.

4.4.3. Objektplausibilisierung nach Schoeberl

Die beiden vorausgegangenen Verfahren haben sich im Wesentlichen darauf konzentriert, die Objektexistenzschätzung basierend auf einer Messfolge mit bestimmten Verteilungs- und Unabhängigkeitsvoraussetzungen zu aktualisieren.

Ein einfaches Verfahren, welches effektiv auch systematische Scheinziele unterdrückt, ist in [Schoeberl 2008] beschrieben. Es setzt eine Monokamera und einen Radarsensor voraus, welche unterschiedliche Messprinzipien haben. Dadurch ist das gleichzeitige Auftreten von systematischen Scheinzielen in den beiden Messfolgen deutlich seltener. Die Messfolgen werden separat ausgewertet, sodass zwei Listen von Objekthypothesen erzeugt werden.

Für jede Objekthypothese wird die Zeit bis zur Kollision mit dem Ego-Fahrzeug (engl. „Time-To-Collision (TTC)“) bestimmt. Diese ergibt sich approximativ aus dem Verhältnis von Abstand d und der Relativgeschwindigkeit v.

Da ein Monokamerasensor weder die Geschwindigkeit noch den Abstand direkt messen kann, muss dieser Parameter anderweitig bestimmt werden. Dieses ist einerseits über die Auswertung des optischen Flusses möglich (vgl. Abschnitt 2.2.1), andererseits über die zeitliche Beobachtung

der Objektkonturgröße.

Bei letzterem Ansatz approximiert der Monokamerasensor für ein Objekt eine Silhouette, wobei bildverarbeitende Algorithmen auf Daten des digitalen Abbildes angewandt werden. Im einfachsten Fall wird hierbei ein Kastenmodell verwendet, sodass nur die Breite und Länge als freie Parameter ermittelt werden müssen.

Es seien zwei aufeinanderfolgende Messungen mit dem zeitlichen Abstand Δt gegeben. Der Abstand des Objektes zum Ego-Fahrzeug sei in den beiden Messungen d_1 bzw. d_2 und dessen Breite b. Es sei ferner angenommen, dass die gemessene scheinbare Objektbreite eines assoziierten Objektes in der ersten und zweiten Messung s_1 bzw. s_2 ist. Für s_1 bzw. s_2 gilt dann:

$$s_1 = \frac{b}{d_1} \tag{4.15}$$

$$s_2 = \frac{b}{d_2} \tag{4.16}$$

Aufgrund der Messdaten der Kamera und der Annahmen lässt sich dann die Zeit bis zur Kollision TTC_K berechnen:

$$\begin{aligned} TTC_K(s_1, s_2) &= \frac{d_2}{v} = \Delta t \frac{d_2}{d_1 - d_2} \\ &= \Delta t \frac{\frac{b}{s_2}}{\frac{b}{s_1} - \frac{b}{s_2}} = \Delta t \frac{s_1}{s_2 - s_1} \end{aligned} \tag{4.17}$$

Um die TTC_R für eine Radarsensormessung zu bestimmen, kann auf bekannte Verfahren zurückgegriffen werden, welche Abstand und Relativgeschwindigkeit zurückliefern [Winner 2009a]. Hierbei ist allerdings die Auswertung zweier rampenförmig modulierter Signale notwendig. Wenn dabei mehrere Objekte auftreten, kann es zu Assoziationsfehlern kommen, sodass eine fehlerhafte TTC_R ermittelt wird.

In [Schoeberl 2008] wird ein verbessertes Verfahren vorgestellt, welches die TTC_R als Quotient aus Abstand und Relativgeschwindigkeit direkt aus einem einzigen empfangenen rampenförmig modulierten Signal bestimmt (vgl. Abschnitt 2.2.3).

Die TTCs in den erzeugten Objektlisten werden miteinander verglichen. Wenn in beiden Objektlisten sich zwei Objekthypothesen mit vergleichbarer TTC finden lassen, so sind diese plausibilisiert und gelten somit als existent.

Das vorgestellte Verfahren ist sehr sensorspezifisch: Mittels spezieller Signalverarbeitung lassen sich Objekthypothesen mit stabiler TTC für Monokamera- und Radarsensordaten generieren, welche als Assoziationskriterium genutzt werden können. Bei einer anderen Sensorplattform könnte dieses Verfahren weniger performant sein.

Insgesamt basiert es auf dem Grundgedanken, dass ein Objekt erst plausibel ist, wenn es von einem zweiten Sensor mit anderem Messprinzip erfasst worden ist. Dieses lässt sich auch in klassischen Trackingverfahren als Plausibilisierungskriterium leicht nutzen. Eine probabilistische Angabe über die Existenzwahrscheinlichkeit bietet dieses Verfahren in dieser Form leider nicht. Sein Fokus liegt im Unterdrücken von Scheinzielen beispielsweise für Assistenzsysteme zur Vermeidung von Kollisionen. Hier sind Fehlauflösungen möglichst zu vermeiden, notfalls auf Kosten fehlender Nichtauslösungen.

4.4.4. Familie der „Probabilistischen Datenassoziationsalgorithmen“

Einer der fortschrittlichsten Ansätze zur Existenzwahrscheinlichkeitsschätzung ist in [Mušicki 2004] beschrieben. Der Schätzer ist Teil eines Objekttrackingalgorithmus und hat in seiner Entstehungsgeschichte verschiedene Einflüsse von anderen Arbeiten geerbt.

Er basiert auf dem Probabilistischen Datenassoziation-Algorithmus (engl. „Probabilistic Data Association (PDA)“) von Bar-Shalom [Bar-Shalom 1975], welcher von Mušicki um den Existenzschätzungsaspekt zum sog. „Integrated Probabilistic Data Association (IPDA)“-Algorithmus erweitert worden ist [Mušicki 1994].

Beide Algorithmen können maximal nur ein einzelnes Ziel verfolgen. Für die Mehrzielfähigkeit wurde der PDA-Algorithmus bereits früh von Bar-Shalom zum „Joint Probabilistic Data Association (JPDA)“-Algorithmus weiterentwickelt [Bar-Shalom 1988]. Eine solche Weiterentwicklung erfolgte auch für den IPDA-Algorithmus durch Mušicki, wodurch letztendlich der sog. „Joint Integrated Probabilistic Data Association (JIPDA)“-Algorithmus [Mušicki 2004] entstand.

Alle Algorithmen der PDA-Familie versuchen im Trackingprozess durch eine probabilistische Darstellung möglichst vieler Aspekte die Anzahl der diskreten Entscheidungen zu minimieren (vgl. Tabelle 4.2). Hierdurch sollen möglichst wenig diskrete Fehlerquellen entstehen. Alle haben vom PDA-Ursprungsalgorithmus eine probabilistische Assoziation zwischen Messungen und Objekthypothesen geerbt.

	Klassisches Tracking	PDA	JPDA	IPDA	JIPDA
Probabilistische Detektion	Nein	Nein	Nein	Ja	Ja
Probabilistische Assoziation	Nein	Ja	Ja	Ja	Ja
Integrierte Existenzschätzung	Nein	Nein	Nein	Ja	Ja
Mehrzielfähigkeit	Ja	Nein	Ja	Nein	Ja

Tabelle 4.2.: Vergleich verschiedener Trackingalgorithmen

Klassische Trackingalgorithmen hingegen nutzen eine Maximum-Likelihood-Funktion, um eine Messung mit einem Track zu assoziieren. Um den Track mit dem geringsten Abstand zur Messung zu finden, wird in vielen Umsetzungen dazu der Mahalanobisabstand verwendet. Ist kein passender Track zu finden, so wird die Messung genutzt, um eine neue Objekthypothese zu initiieren. Dieser einfache Trackingalgorithmus wird auch „Global Nearest Neighbour (GNN)“-Algorithmus genannt.

Hierbei treten insgesamt drei verschiedene Fehlerquellen hervor: Zum einen können Tracks initialisiert werden, die auf Scheinquellen oder Messrauschen basieren, wobei sog. „False Positives“ entstehen. Zudem können Messungen zu falschen Tracks zugeordnet werden, wobei sog. Assoziationsfehler auftreten, welche Objekte mit falschen Messdaten aktualisiert werden. Als letzte Fehlerquelle könnte eine Objekthypothese fälschlicherweise frühzeitig verworfen werden, da das zugehörige Objekt beispielsweise durch Verdeckung kurzfristig nicht detektiert werden konnte.

Der PDA-Algorithmus im ursprünglichen Sinn ist für Umgebungen optimiert, in der ein einzelnes Objekt zusammen mit vielen Scheinobjekten auftritt. Hierbei kann es durch Fehlassoziation leicht zu einem Trackverlust kommen. Deshalb berechnet der PDA-Algorithmus für alle benachbarten Messungen eines Tracks die Wahrscheinlichkeit, dass diese Messungen zu diesem Track gehören. Der Track selbst wird anschließend durch alle benachbarten Messungen aktualisiert, welche mit der Assoziationswahrscheinlichkeit gewichtet werden. Der Algorithmus nimmt dabei alle Modelle als linear bzw. Zufallsvariablen als normalverteilt an.

In vielen Umgebungen sind allerdings häufig mehrere reale Ziele zu finden. Dafür wurde der PDA-Algorithmus zum „JPDA“ erweitert, der mehrzielfähig ist. Hierbei muss insbesondere der Fall beachtet werden, dass,

wenn eine Messung im Bereich mehrerer Tracks liegt, diese auch zu mehreren Tracks assoziiert werden kann. Die Existenz aller Tracks gilt als sicher.

Die Erweiterung des PDA-Algorithmus um den Existenzaspekt erfolgt über die bayessche Schätzung der Eintrittswahrscheinlichkeit eines zusätzlichen Ereignisses. Dieses Ereignis betrifft das Vorhandensein und Beobachtbarkeit bzw. Nichtvorhandensein eines einzelnen Objektes. IPDA nimmt dabei eine gleichverteilte Falschalarmrate im beobachteten Bereich an.

Um den IPDA-Algorithmus um den Aspekt der Mehrzielfähigkeit und somit zum JIPDA-Algorithmus zu erweitern, ist in der Praxis eine komplexere Trackverwaltung notwendig. Eine große Herausforderung ist dabei, dass im Laufe des Trackingprozesses sehr viele Tracks mit zum Teil geringer Existenzwahrscheinlichkeit entstehen. Die Anzahl möglicher Tracks steigt durch die probabilistische Assoziation und Existenzschätzung während der rekursiven Objektschätzung exponentiell mit der Anzahl der Schritte. Intelligente Strategien sind notwendig, um den entstehenden Hypothesenbaum auf vielversprechende Zweige zu beschneiden (vgl. [Mählisch 2009; Munz 2011]).

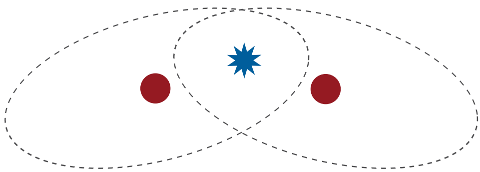


Abbildung 4.2.: Zwei Objekthypothesen (rote Punkte) konkurrieren um eine Messung (blauer Stern), sodass die Existenzmasse aufgeteilt wird. Die Aufteilung folgt nach Gewichtung von Mahalanobisabstand (Ellipsen) und aktueller Existenzwahrscheinlichkeit der Tracks.

Das Beschneiden des Hypothesenbaumes hat Auswirkungen auf die verbleibenden Tracks. Falls beispielsweise die Existenzmasse einer Messung auf zwei konkurrierende Objekthypothesen aufgeteilt worden ist, und eine dieser beiden Hypothesen im Laufe des Trackings später verworfen wird, so erhält der verbleibende Track dessen Existenzmasse (s. Abbildung 4.2). Um diese genau zu bestimmen, muss die Fehlentscheidung korrigiert werden, die Glaubensmasse auf zwei Tracks aufzuteilen. Letztendlich müssen alle betroffenen Knoten in der Baumstruktur bis zu den

Blättern aktualisiert werden. Die Blätter des Baumes repräsentieren die aktuelle Schätzhypothese.

Der JIPDA-Algorithmus vereint alle Aspekte der probabilistischen PDA-Algorithmen. Im Vergleich zu anderen Trackingalgorithmen müssen im gesamten Algorithmus theoretisch keine diskreten Entscheidungen mehr gefällt werden. In der Praxis ist eine Begrenzung der exponentiell steigenden Trackhypothesenanzahl jedoch notwendig, sodass bei Erreichung bestimmter Existenzwahrscheinlichkeitsschwellwerte oder beim Überschreiten einer maximalen Anzahl von Trackhypothesen diese nicht weiter verfolgt werden.

Nachteilig ist jedoch, dass keine systematischen Störungen berücksichtigt werden. So wird auch hier eine gleichverteilte Fehlzielrate angenommen. Ebenso finden sich die üblichen Annahmen zu linearen Modellen und normalverteilten Zufallsvariablen wieder.

4.4.5. Entwicklung eines Existenzwahrscheinlichkeitsschätzer mithilfe von zyklischen Markov-Ketten

Der folgende Existenzwahrscheinlichkeitsschätzer verwendet zyklische Markov-Ketten, um Messabhängigkeiten zu modellieren. Zur Bestimmung der Existenzwahrscheinlichkeit nutzt er allein die Ereignisfolge M^k eines Tracks. Diese gibt an, ob eine Messung erfolgreich zu diesem Track zugeordnet werden konnte oder nicht. Diese besteht aus n Einzelereignissen mit $m_i \in M$ und $m_i = \{+,-\}$. „$+$" bedeutet, dass in einem Updateschritt mindestens eine Messung erfolgreich mit dem Track assoziiert werden konnte, während bei dem Ereignis „-" dies bei keiner Messung möglich war.

Die Markov-Kette hat zwei Zustände und somit ein Gedächtnis, ob beim letzten Updateschritt die Assoziation erfolgreich war (Zustand A) oder nicht erfolgreich war (Zustand $\overline{A}$). Insgesamt werden zwei Markov-Ketten verwendet. Die eine repräsentiert die Zustandsübergänge für die Situation, dass das Objekt existiert (H_1), während die zweite Markov-Kette von einem nicht existenten Objekt (H_0) ausgeht. Die beiden Markov-Ketten und ihre Transitionswahrscheinlichkeiten sind in Abbildung 4.3 dokumentiert.

Die Markov-Ketten lassen sich durch vier Parameter beschreiben, welche die Persistenzwahrscheinlichkeiten für die jeweiligen Zustände der beiden Markov-Ketten repräsentieren. Diese geben die jeweilige Wahrscheinlichkeit an, ob im nächsten Traversierungsschritt die Markov-Kette in dem aktuellen Zustand verharrt. Zusätzlich zu den Persistenzwahrscheinlichkei-

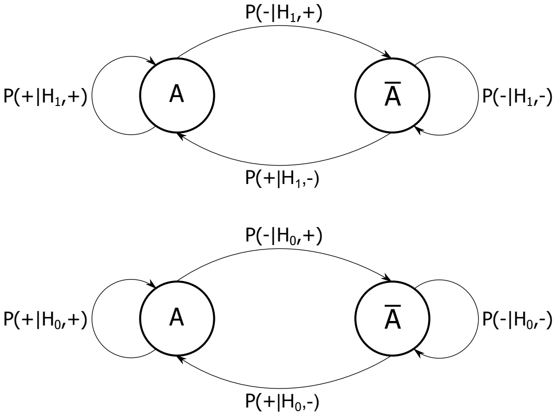


Abbildung 4.3.: Zwei zyklische Markov-Ketten: Die obere modelliert die Situation, dass ein Objekt existiert (H_1) während die untere nichtexistente Objekte (H_0) beschreibt.

ten ist ein Parameter für die initiale Existenzwahrscheinlichkeit notwendig. Eine Beschreibung der Parameter ist aus Tabelle 4.3 zu entnehmen.

Der Wechsel des Zustandes innerhalb einer der beiden Markov-Ketten wird durch das entsprechende Gegenereignis modelliert. Deren Wahrscheinlichkeit kann komplementär berechnet werden. Für die erste Markov-Kette gilt für die Zustandswechselwahrscheinlichkeiten:

$$P(-|H_1,+) \quad = \quad 1-P(+|H_1,+) \tag{4.18}$$
$$P(+|H_1,-) \quad = \quad 1-P(-|H_1,-) \tag{4.19}$$

Gleiches gilt entsprechend für die zweite Markov-Kette. Aus diesen Konstanten lassen sich die Aufenthaltswahrscheinlichkeiten $\Omega_{H_1}(A)$ bzw. $\Omega_{H_0}(A)$ ermitteln. Diese beschreiben die relative Häufigkeit für repräsentative Messfolgen, dass bei der Traversierung der Markov-Ketten der Zustand A eingenommen wird:

$$\Omega_{H_1}(A) \quad = \quad \left(\frac{P(+|H_1,-)}{P(-|H_1,+)}+1\right)^{-1} \tag{4.20}$$
$$\Omega_{H_0}(A) \quad = \quad \left(\frac{P(+|H_0,-)}{P(-|H_0,+)}+1\right)^{-1} \tag{4.21}$$

Ist eine Messfolge M^k gegeben, so lassen sich die relativen Wahrscheinlichkeiten bestimmen, ob diese bei einem existenten Objekt $P(M^k|H_1)$ oder nichtexistenten Objekt $P(M^k|H_0)$ auftritt. Hierbei ist zu beachten, dass eine Messfolge immer mit einer Detektion „+“ beginnt, da ohne eine Erstdetektion keine Trackinitialisierung stattfindet. Somit startet die Markov-Kette im Zustand A mit der a-priori bestimmten Aufenthaltswahrscheinlichkeit $\Omega_A(H_1)$ bzw. $\Omega_A(H_0)$. Anschließend werden die Markov-Ketten anhand der Messfolge traversiert, wobei die Kantengewichte multiplikativ in die Wahrscheinlichkeitsberechnung einfließen:

$$P(M^k|H_1) = \Omega_A(H_1) \cdot \prod_{i=2}^{k} P(m_{i-1}|H_1, m_i) \quad (4.22)$$

$$P(M^k|H_0) = \Omega_A(H_0) \cdot \prod_{i=2}^{k} P(m_{i-1}|H_0, m_i) \quad (4.23)$$

Je länger die Messfolge ist, desto geringer sind die relativen Wahrscheinlichkeiten $P(M^k|H_1)$ bzw. $P(M^k|H_0)$. Ein sinnvolles Maß, die diese Wahrscheinlichkeiten in Bezug setzt, ist die „Likelihood-Ratio“ $LR(M^k)$. Dieses wird beispielsweise auch in [Sittler 1964] verwendet und ergibt sich aus dem Quotienten der relativen Wahrscheinlichkeiten:

$$LR(M^k) = \frac{P(M^k|H_1)}{P(M^k|H_0)} \quad (4.24)$$

$$= \frac{P(M^k|H_1)}{1 - P(M^k|H_1)} \quad (4.25)$$

Die $LR(M^k)$ lässt sich mittels Bayes so darstellen, dass der Term ausschließlich die Initialwahrscheinlichkeiten $P_0(H_1)$ bzw. $P_0(H_0)$ und die Existenz- bzw. Nichtexistenzwahrscheinlichkeit beinhaltet. Die Detektionswahrscheinlichkeit $P(M^k)$ kürzt sich dabei heraus und muss somit nicht parametrisiert werden:

$$LR(M^k) = \frac{P(M^k|H_1)}{P(M^k|H_0)} \quad (4.26)$$

$$= \frac{P_0(H_1)}{P_0(H_0)} \cdot \frac{P(H_1|M^k)}{P(H_0|M^k)} \quad (4.27)$$

Mittels der Gleichungen 4.25 und 4.27 lässt sich aus der Likelihood-Ratio und den A-priori-Wahrscheinlichkeiten eine Existenzwahrscheinlichkeit ermitteln:

$$P(H_1|M^k) = \left(1 + \frac{P_0(H_0)}{P_0(H_1)} LR(M^k)^{-1}\right)^{-1} \tag{4.28}$$

Parameter	Beschreibung
$P_0(H_1)$	Initiale Existenzwahrscheinlichkeit einer Objekthypothese
$P(+\|H_1,+)$	Persistenzwahrscheinlichkeit eines existierenden Objekts („Persistent True Positive“)
$P(-\|H_1,-)$	Persistenzwahrscheinlichkeit einer Nichtdetektion eines existierenden Tracks („Persistent False Negative“)
$P(+\|H_0,+)$	Persistenzwahrscheinlichkeit eines Scheinziels („Persistent False Positive“)
$P(-\|H_0,-)$	Persistenzwahrscheinlichkeit einer Nichtdetektion eines nichtexistierenden Objekts („Persistent True Negative“)

Tabelle 4.3.: Parametrisierbare Attribute für den auf einen zyklischen Markov-Ketten-Ansatz beruhenden Existenzschätzers

Für die Parametrisierung dieses Schätzers müssen die 5 Attribute aus Tabelle 4.3 bestimmt werden. Hierzu sind Ground Truth-Daten[2] notwendig, aus denen sich leicht die initiale Trackexistenzwahrscheinlichkeit $P(H_1)$ ermitteln lässt.

Um die Persistenz- bzw. Übergangswahrscheinlichkeiten zu bestimmen, muss mit den Ground Truth-Daten die jeweils zugehörige Markov-Kette traversiert werden. Jedes Mal wenn eine Kante besucht wird, erhöht sich deren Gewicht um eins. Nach Abschluss aller Traversierungen werden die Kanten normiert, sodass die Summe aller Kantengewichte die einen Knoten verlassen, eins ergibt.

Das vorgestellte Verfahren ist das bisher einzige, welches systematische Fehlziele berücksichtigt, da es die Persistenzwahrscheinlichkeit eines Fehlziels explizit als Parameter nutzt. Es ist leicht zu implementieren und parametrisieren, weshalb es in der im vorherigen Kapitel vorgestellten Sensordatenfusion intern als Gütemaß Verwendung findet.

Nachteilig ist, dass es als einzigen Faktor nur das Ereignis der erfolgreichen Assoziation einer Messung zu einem Track berücksichtigt und somit viele andere mögliche Parameter außer Acht lässt. Es ist fraglich, ob dieser

[2] „Ground Truth“-Daten sind Referenzdaten, die auf Messdaten basieren. So kann ein Experte beispielsweise Ground Truth-Daten für Existenzschätzer erzeugen, indem er Messdaten manuell entsprechend klassifiziert.

Klassifikator mit so wenig Informationen den Zustandsraum so gut trennen kann, dass er nach kurzer Zeit Scheinziele von existenten Zielen sicher unterscheidet.

Nachteilig ist außerdem, dass für verschieden lange Messfolgen keine unterschiedlichen Markov-Ketten existieren. So scheint es wahrscheinlicher, dass je länger ein Objekt getrackt werden kann, dieses auch stabil persistent gemessen werden kann, da es ansonsten früher oder später verworfen werden würde. Dieser einzelne Track würde aber als Ground Truth-Track die Persistenzwahrscheinlichkeit für existierende Objekte massiv ändern, welche nur durch sehr viele kurze Tracks mit Scheinzielen wieder ausgeglichen werden könnte. Die Kantengewichte bilden somit ein Mittel über alle Assoziationen, sodass das Ergebnis wohl nur statistisch korrekt ist, wenn der zu untersuchende Track genau die mittlere Trackdauer aufweist.

Das gleiche Phänomen der Mittelung tritt bei der Betrachtung einzelner Kantengewichte auf. So wird beispielsweise die Persistenzwahrscheinlichkeit von Scheinzielen über alle Störphänomene in den Trainingsdaten gemittelt, ohne dass diese Wahrscheinlichkeit beispielsweise anhand von Messfolgelängen differenziert wird.

4.4.6. Statische Regeln

Ein weitverbreiteter Ansatz zur diskreten Existenzbestimmung ist die Verwendung statischer Regeln, die die Objekthypothesen erfüllen müssen, um als existent betrachtet werden zu können. Während des Objekttrackings ist jede dieser Hypothesen durch verschiedene Parameter gekennzeichnet, die als Faktoren zur Existenzklassifikation genutzt werden können. Diese werden in einem Regelkatalog in Form von verknüpften booleschen Ausdrücken ausgewertet. Hierbei müssen diese Faktoren sorgsam gewählte Schwellwerte über- oder unterschreiten, damit die Objekthypothese als existent gelten darf.

Als ein Teilfaktor für die Existenzbetrachtung wird bei vielen Assistenzsystemen die Dynamik der Objekthypothese ausgewertet. Dabei werden häufig stationäre Objekte herausgefiltert, was die überwiegend meist stationären Scheinziele eliminiert. Um Fehlauslösungen durch solche Scheinziele zu vermeiden, ist bei vielen Systemen die Nichtbetrachtung stationärer Objekte Teil der Spezifikation geworden. So betont die ISO-Norm zum Abstandsregeltempomaten [ISO 2002], dass die Betrachtung solcher Objekte keine Anforderung ist und dass dieser Umstand nur geeignet im Handbuch dokumentiert werden muss. Neuere ACC-Systeme („Full Speed

Range Adaptive Cruise Control“) beachten zumindest noch stationäre Objekte, die sich im Laufe des Trackings bewegt hatten [Winner 2009b]. Auf diese kann die Geschwindigkeit des Ego-Fahrzeuges dann bis zum Stillstand geregelt werden.

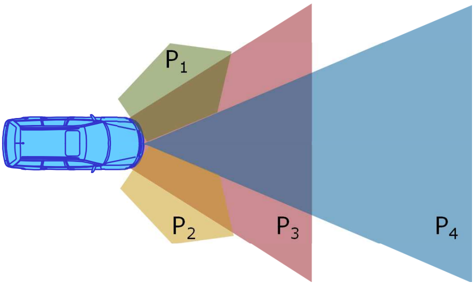


Abbildung 4.4.: Objekthypothesen müssen je nach Position unterschiedliche Bedingungen erfüllen. Die Gültigkeitsbereiche dieser Bedingungen sind durch Polygone dargestellt (vgl. [Effertz 2009, S. 98 ff]).

Als weitere statische Regel kann die Plausibilisierung durch mehrere Sensoren genutzt werden, wie in Abschnitt 4.4.3 exemplarisch beschrieben worden ist. In [Effertz 2009, S. 98 ff] ist der Ansatz durch die Definition von Polygonen verfeinert worden, die fest im Fahrzeugkoordinatensystem liegen und mit den Sichtbereichen der Umfeldsensoren korrelieren. Vier solcher Polygone sind beispielhaft in Abbildung 4.4 dargestellt. Für jedes dieser Polygone wird ein boolescher Ausdruck definiert, der eine Stützung einer Objekthypothese durch einzelne Sensoren fordert oder auch ausschließt. Nur wenn dieser Ausdruck je nach Konfiguration ein- oder mehrfach während des Objekttrackings erfüllt worden ist, gilt die zugehörige Objekthypothese als existent. Durch eine solche Polygondefinition wird die Position der Objekthypothese berücksichtigt. Um sinnvolle Bedingungen aufstellen zu können, erfordert dieser Ansatz jedoch ein hohes Maß technologisch möglichst heterogener Sensoren mit redundanten Detektionsbereichen.

Als weiteres Kriterium kann eine minimale Trackingdauer festgelegt werden. Der Unterschied zum Kriterium der Mindestzahl von Messassoziationen ist, dass der Track mit wenigen aber zeitlich gleichmäßig verteilten Messungen plausibilisiert werden kann. Dies kann unter Umständen vor systematischen Fehlzielen schützen, da sich existente Objekte wäh-

rend der minimalen Trackingdauer komplett im Sichtbereich befinden müssen. So kann beispielsweise eine Schilderbrücke aus dem Erfassungsbereich eines Radarsensors und Laserscanners verschwinden, wenn sich das Fahrzeug dieser genügend angenähert hat. Geschieht dies innerhalb der minimalen Trackingdauer, so wird die Objekthypothese zu keinem Zeitpunkt als existent angesehen.

Weitere denkbare Regeln können Höchstunsicherheiten von einzelnen Zustandsvarianzen betreffen oder deren Kombination in Form der Determinante der Kovarianzmatrix, wie sie Sittler (vgl. Abschnitt 4.4.1) und Sinha (vgl. Abschnitt 4.4.2) auch nutzen. Des Weiteren können die Ausgaben anderer Existenzwahrscheinlichkeitsschätzer in den Regelkatalog einfließen, indem diese beispielsweise bestimmte Schwellwerte über- bzw. unterschreiten müssen, bevor ein Objekt als existent bzw. nichtexistent angesehen wird.

Die Vorteile des Verfahrens der statischen Regeln liegen in der hohen Transparenz und schnellen Anpassbarkeit. So lassen sich praktisch alle Regeln leicht parametrisierbar in Software implementieren. Die Adaption der Parametrisierung an die Sensorik, an das Assistenzsystem und dessen Umgebung kann dann schnell erfolgen.

Nachteilig an dem Verfahren ist, dass komplexere Zusammenhänge eventuell nur schwer von einem Experten zu erfassen sind, um daraus korrekte Regeln abzuleiten. Häufig werden deshalb einfache Schwellwertregeln erstellt, die mit „UND“ bzw. „ODER“ verknüpft werden. Die Trennmöglichkeit solcher Funktionen für komplexere Zustandsräume ist allerdings begrenzt, sodass diese eventuell eine schlechtere Performance aufweisen als andere Verfahren. Die Regeln sind zudem stark abhängig vom Typ des Assistenzsystems und der Sensorik, sodass die Schwellen für Existenz bzw. Nichtexistenz bereits implizit in den Regeln integriert werden müssen.

Auch ist bei der Festlegung der Schwellwertparameter für die einzelnen Regeln häufig eine unsystematische evolutionäre Entwicklung festzustellen. Hierbei wird durch mehrstufige Parametervariation versucht, die Raten von Fehlauslösungen und fehlerhaften Nichtauslösungen zu optimieren. Je nach Komplexität des Klassifikationsproblemes kann diese Vorgehensweise sehr aufwendig und uneffizient sein.

4.4.7. Bewertung

Das Unterdrücken nichtexistenter Ziele ist in vielen wissenschaftlichen Arbeiten bereits früh untersucht worden. Der vorwiegende Anwendungsbereich der meisten Arbeiten liegt jedoch im Luftfahrtbereich und hier ins-

besondere bei der Überwachung von Lufträumen mittels Radaranlagen im militärischen Kontext. Alle Verfahren sind somit geeignet, sehr kurze Scheinziele zu unterdrücken, die aus Messrauschen stammen. Besonders positiv ist bei JIPDA zudem die Integration in einem Trackingalgorithmus zu erwähnen, sodass der Zustand einzelner Trackhypothesen durch solche Scheinziele möglichst wenig negativ beeinflusst wird.

Im Vergleich zur Fahrzeugumfelddomäne ist die Domäne des Luftraumes weniger komplex strukturiert. Wie bereits erwähnt, ist im Fahrzeugumfeld mit Artefakten zu rechnen, die systematische Scheinziele verursachen. Auch tritt eine Vielfalt von existenten Objekttypen auf, die alle von den systematischen Scheinzielen unterschieden werden müssen. Des Weiteren spielen Verdeckungen hier eine größere Rolle.

	Sittler	Sinha	Schoeberl	JIPDA	zykl. Markov-Ketten	statische Regeln
Probabilistische Schätzung	•	•		•	•	
Berücksichtigung systematischer Scheinziele			•		•	•
Berücksichtigung einer heterogenen Sensorkonfiguration			•	•		•
Berücksichtigung vieler Attribute	•	•		•		•
Existenz als Klassifikationsproblem			•			•

Tabelle 4.4.: Vergleich verschiedener Existenzschätzer

Dementsprechend sind viele der vorgestellten Verfahren für eine Existenzschätzung im Sinne von Definition 1 weniger geeignet. Dieses zeigt sich insbesondere dann, wenn diese für Messfolgen bestimmte Unabhängigkeitsannahmen machen, wie beispielsweise gleichverteilte Scheinzielraten. Eine Ausnahme bildet die Existenzschätzung mittels zyklischer Markov-Ketten, welche explizit die Persistenzwahrscheinlichkeiten modellieren.

Zu kritisieren ist auch, dass viele Schätzer hauptsächlich das Ereignis der Trackassoziation bzw. Nichtassoziation nutzen, um ein Existenzgütemaß zu ermitteln. Dies hat zur Folge, dass diese vielmehr ein „Beobachtbarkeitsgütemaß“ anstatt eines Existenzmaßes liefern: So verringern viele Schätzer beim Ereignis der Nichtassoziation dessen Gütemaß, auch wenn der Track in der Vergangenheit bereits zweifelsohne als existenter Track klassifiziert worden ist. Wird ein Objekt verdeckt oder verlässt ein Objekt den Sichtbereich der Sensoren, so wird die Existenzwahrscheinlichkeit kontinuierlich verringert, bis das Objekt als nichtexistent angesehen wird. Dennoch kann bezweifelt werden, dass das Objekt aufgehört hat zu existieren, nur weil es nicht mehr wahrgenommen werden kann.

Nachteilig an vielen Verfahren ist außerdem die Erfordernis, dass alle Zustandsunsicherheiten genau bestimmt werden müssen, um eine gute Existenzschätzung treffen zu können. Da insbesondere bei einer heterogenen Sensorplattform sehr viele Faktoren zu berücksichtigen sind, ist eine manuelle Modellierung sehr herausfordernd. Außerdem werden teilweise unrealistische Modellannahmen gemacht, sodass es fraglich ist, ob eine statistisch korrekte Existenzwahrscheinlichkeit[3] in der Realität durch einen Schätzer tatsächlich geliefert wird. Hierbei ist insbesondere zu kritisieren, dass keine einzige bekannte Arbeit anhand von realen Mess- und Ground-True-Daten den jeweiligen Schätzer probabilistisch validiert hat.

4.5. Stand der Technik: Signifikanzschätzer

Wenn ein Objekt als existent angesehen wird, muss noch bestimmt werden, ob es im Sinne des Fahrerassistenzsystems signifikant ist (vgl. Definition 2 im Abschnitt 4.3). Die Signifikanzanalyse ist im hohen Maße abhängig vom Typ des Assistenzsystems: Für einen Spurwechselassistenten sind beispielsweise nur Objekte signifikant, die sich in Nachbarfahrstreifen befinden. Für Assistenzsysteme zur Vermeidung von Kollisionen sind nur Objekte wesentlich, die sich vor dem Ego-Fahrzeug befinden und bei denen eine geringe TTC gegeben ist.

Exemplarisch werden im Folgenden bekannte Ansätze für einen Abstandsregeltempomaten (ACC) vorgestellt. Dabei wird ein Korridor, der sog. Fahrschlauch definiert, in dem Objekte potenziell als signifikant klassifiziert werden. Diese können dann den Regelkreis der Längsaktorik maßgeblich beeinflussen. Dieser Korridor ist durch einen Kursverlauf und eine konstante Breite oder durch eine distanzabhängige Breitefunktion definiert.

[3] Vergleiche hierzu die Definition von statistisch korrekten Klassifikatoren im Anhang A.1.2.

Fehler in der Kursverlaufs- und Fahrschlauchbreitenschätzung können sich negativ auf die Signifikanzanalyse und somit negativ auf das Fahrerassistenzsystem auswirken. Dieses kann zu unnötigen Reaktionen auf scheinbare Objekte in der Längsregelung führen bzw. eine notwendige Reaktion auf relevante Objekte unterlassen. In beiden Fällen muss der Fahrer das System überstimmen und übersteuern. Geschieht dies zu häufig, kann das den Komfort und die Sicherheit mindern und somit die Akzeptanz für dieses System verringern.

Verschiedene diskrete Ansätze zur Bestimmung des Kursverlaufes und der Fahrschlauchbreite werden in den beiden nachfolgenden Abschnitten beschrieben. Um ein Gütemaß der Signifikanzschätzung zu ermitteln und somit die Signifikanzanalyse zu verbessern, werden anschließend probabilistische Erweiterungen zu diesen diskreten Schätzern vorgestellt und bewertet.

4.5.1. Bestimmung des Kursverlaufes

Viele ACC-Systeme schätzen den zukünftigen Kursverlauf anhand der Krümmung der Fahrbahn. Da das Ego-Fahrzeug auf dieser fährt, kann approximativ dessen Odometriesensorik eingesetzt werden, um die Krümmung auf der Höhe des Ego-Fahrzeuges zu bestimmen. Hierzu bieten sich Daten aus dem Gierratensensor, dem Querbeschleunigungssensor, dem Lenkradwinkelsensor und den Raddrehzahlsensoren an (s. Kapitel 2 und [Winner 2009b]).

Eine solche Krümmungsbestimmung allein erlaubt ausschließlich eine Fahrschlauchgenerierung in Kreisform. Durch die Ableitung der Krümmung über die zurückgelegte Strecke ist zusätzlich eine Approximation der Krümmungsänderung möglich, welche die Erweiterung des Kreismodells auf ein Klothoidenmodell ermöglicht.

Dieses Vorgehen ist von der Kostenseite her sehr attraktiv, da die Odometriesensoren meist bereits im Fahrzeug vorhanden sind. Eine weitere bereits vorhandene Datenquelle sind stationäre Ziele (z. B. Leitplanken), die aus dem ACC-Umfeldsensor gewonnen werden können. Diese begrenzen häufig den Fahrbahnrand und liefern somit einen Anhaltspunkt über den zukünftigen Fahrbahnverlauf. Auch die Mittelung der Quergeschwindigkeiten von mehreren vorausfahrenden Objekten gibt einen Hinweis über die zukünftige Fahrbahnkrümmung.

Nichtsdestotrotz wird die Fahrbahngeometrie nur indirekt bestimmt, wobei diese indirekten Quellen zwar häufig aber nicht immer mit dem Fahrbahnverlauf korrelieren. So können sich ohne direkte Informationsquellen

über den Fahrbahnverlauf in bestimmten Situationen Doppeldeutigkeiten ergeben, welche eine sichere Signifikanzbestimmung beeinträchtigen.

Als Beispiel sind dazu zwei identische Objektkonstellationen in Abbildung 4.5 bei unterschiedlichen Fahrbahnen dargestellt. Diese indirekte Informationsquellen suggerieren eine Kurve, jedoch ist auch eine gerade Straße möglich. Je nach Fahrbahnhypothese können sich somit unterschiedliche Objektsignifikanzen einstellen.

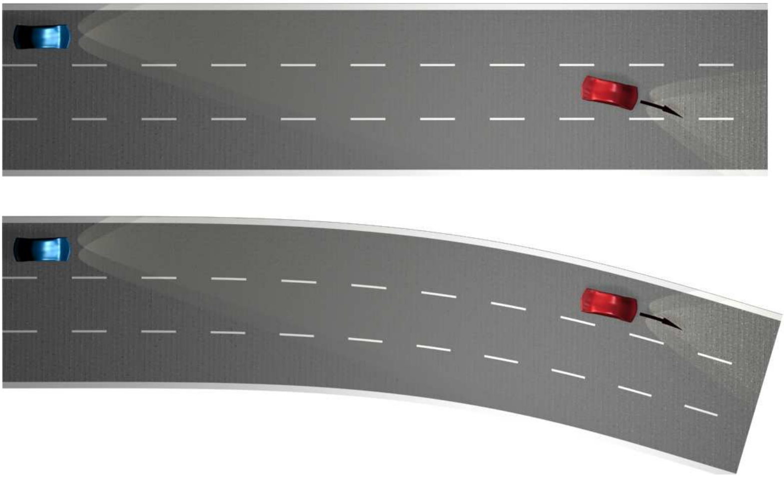


Abbildung 4.5.: Gleiche Objektkonstellationen weisen bei unterschiedlichen Fahrbahnen unterschiedliche Signifikanzen auf (vgl. [Winner 2009b]).

Um solche Fehleinschätzungen zu vermeiden und eine genauere Bestimmung des Kursverlaufes zu ermöglichen, ist die Fusionierung direkter Datenquellen, wie digitaler Karte und Fahrstreifenmarkierungen aus einem Kamerasensor notwendig. Für eine exemplarische Beschreibung einer solchen Teilfusion sei dazu auf Abschnitt 3.5.5 verwiesen.

4.5.2. Bestimmung der Fahrschlauchbreite

Der Fahrschlauch wird aus dem prädizierten Kurs und einer Fahrschlauchbreite erzeugt. Letztere kann konstant oder auch variabel über der Strecke sein, sodass der Fahrschlauch komplexere Formen annehmen kann.

Bei der Festlegung der Fahrschlauchbreite tritt ein Zielkonflikt auf: Wird dieser zu groß gewählt, so reagiert das ACC-System evtl. auf nicht signifikante Objekte. Umgekehrt können bei einer zu kleinen Fahrschlauchbreite fälschlicherweise Objekte ignoriert werden, welche sich auf dem Ego-Fahrstreifen befinden.

Eine variable Fahrschlauchbreite kann die Wahrscheinlichkeit von Falschzuordnungen verkleinern. So ist eine Aufweitung des Fahrschlauches nach links bzw. rechts sinnvoll, wenn dort keine befahrbaren Nachbarfahrstreifen sind. Auf das Vorhandensein bzw. Nichtvorhandensein solcher Nachbarfahrstreifen kann aus der Beobachtung von Fremdobjekten bzw. stationären Randobjekten geschlossen werden (vgl. [Urban 2006]).

Auch die Beobachtung von Fahrstreifenmarkierungsarten mithilfe eines Kamerasensors kann einen Hinweis auf das Vorhandensein von Nachbarfahrstreifen geben (vgl. [Weiser 2011]). Als weitere Informationsquelle kann der Straßentyp dienen, der beispielsweise aus einer digitalen Karte entnommen werden kann. So ist auf Autobahnen, gut ausgebauten Landstraßen und auf wichtigen Verbindungsstraßen im innerstädtischen Bereich mit mehreren Fahrstreifen pro Richtung zu rechnen, während auf den meisten anderen Straßen nur ein Fahrstreifen pro Richtung zur Verfügung steht.

4.5.3. Probabilistische Fahrschlauchgenerierung und -zuordnung

Im einfachsten Fall kann direkt aus der Fahrschlauchbreite und dem prädizierten Kursverlauf der Fahrschlauch generiert werden. In [Winner 2009b] wird außerdem vorgeschlagen, dem Fahrschlauch eine unscharfe Kontur zu geben, sodass eine probabilistische Aussage möglich ist, inwieweit sich ein Objekt in diesem befindet. Abbildung 4.6 zeigt diesen Ansatz exemplarisch. Zu beobachten ist, dass sich die Signifikanzdichte in der Entfernung verringert, sodass die tendenziell ungenaue Fahrbahnschätzung in großer Entfernung zum Tragen kommt.

In [Urban 2006] werden weitere Faktoren bestimmt, um das probabilistische Fahrschlauchmodell zu verfeinern. Hierzu gehören die Wahrscheinlichkeiten, ob sich links bzw. rechts vom Ego-Fahrstreifen Fahrbahnbegrenzungen befinden. Auch werden Wahrscheinlichkeiten bestimmt, ob Parallelspuren beispielsweise durch überholende Fahrzeuge erkannt worden sind. Nachgeschaltete Tiefpassfilter mitteln die bestimmten Einzelwahrscheinlichkeiten, um ein konsistenteres Umgebungsmodell zu erhalten.

Ebenso wird ein Tiefpassfilter dazu verwendet, die Objektzuordnungsgüte stabiler zu schätzen. Dadurch kann eine kurzfristige Fehleinschätzung

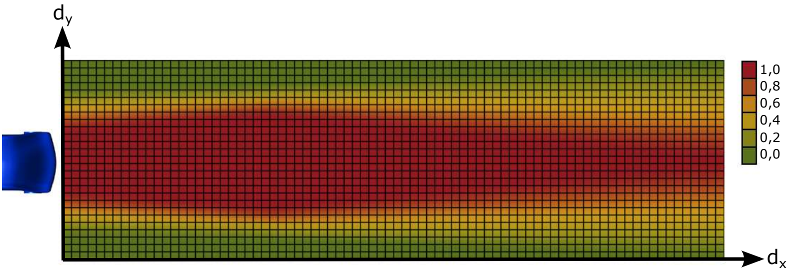


Abbildung 4.6.: Unscharfe Kontur zur Signifikanzmodellierung (vgl. [Winner 2009b])

des Kurses oder der Objektposition kompensiert werden, ohne dass das Regelobjekt verloren geht.

Um zu vermeiden, dass Objekte bei einer geringfügigen Schwankung der Objektzuordnungsgüte verworfen und kurze Zeit später neu initialisiert werden, ist die Nutzung einer Hysteresefunktion vorteilhaft. Um ein Objekt erstmalig als signifikant zu klassifizieren, muss dieses einen höheren Gütewert erreichen als beim Verwerfen der Signifikanzhypothese.

4.5.4. Bewertung

Viele eingesetzte Signifikanzschätzer für ACC-Systeme nutzen sehr wenige Daten aus: Häufig werden nur die Daten eines Radarsensors mit den Ego-Fahrzeug-Odometriedaten verknüpft. Die Fusionierung und Ausgabe der Regelgrößen erfolgt meist gleich im Steuergerät des Sensors, sodass die adaptive Geschwindigkeitsregelung mit nur einer zusätzlichen Hardware umgesetzt werden kann.

Wie vorgestellt, gibt es weitere Ergänzungen durch digitale Karten und Kameradaten, welche die Entscheidungsgrundlage verbreitern und eine genauere Signifikanzbestimmung ermöglichen.

Alle vorgestellten Verfahren verwenden statische Regeln oder nur sehr rudimentäre probabilistische Verfahren, um die Signifikanz zu bewerten. So gibt es beispielsweise in [Winner 2009b] weder eine Angabe darüber, wie man die unscharfe Fahrschlauchkontur festlegen soll, noch wie man verschiedene Signifikanzwerte über die Zeit fusioniert. Noch schwieriger wird es, alle vorgeschlagenen probabilistischen Quellen zu einer statistisch korrekten Aussage zu fusionieren, da diese Quellen nicht als unabhängig angenommen werden können.

4.6. Entwicklung eines integrierten Relevanzschätzer

Um die Objektrelevanz zu bestimmen, nutzen klassische Verfahren die indirekte Herleitung über die in den vorherigen Abschnitten aufgeführten Existenz- und Signifikanzwahrscheinlichkeiten. Relevant ist ein Objekt genau dann, wenn beide Schätzer dies bestätigen.

Die analytische Modellbildung der Objektexistenz und -signifikanz ist jedoch durch Vereinfachungen und Annahmen der statistischen Abhängigkeit bzw. Unabhängigkeit geprägt. Die Berücksichtigung von in der Umwelt vorkommenden Artefakten, die die Umfeldwahrnehmung negativ beeinflussen, findet kaum oder gar nicht statt. Die einfache Kombination der Objektexistenz und -signifikanz erzeugt zudem einen Klassifikator, der davon ausgeht, dass beide Schätzergebnisse unabhängig voneinander sind.

In diesem Abschnitt wird alternativ ein Klassifikator vorgestellt, der die Klasse der relevanten von der Klasse der irrelevanten Objekte direkt im Sinne einer spezifischen Fahrerassistenzapplikation zu separieren vermag. Durch diese Klassenreduzierung erscheint es potenziell möglich, diesen leistungsfähiger zu gestalten.

Die analytische Modellerstellung für einen direkten Relevanzschätzer erscheint aufgrund der hochgradig unbekannten inneren Struktur des Gesamtsystems sehr komplex: Hier müssen beispielsweise die Auftretenswahrscheinlichkeiten von systematischen Störphänomenen berücksichtigt werden, die statistischen Abhängigkeiten verschiedener Informationsquellen sowie deren Ungenauigkeiten. Möchte man deshalb auf die manuelle Modellierung dieser Klassifikationsfunktion verzichten, so bieten sich Lernverfahren zu deren Approximation an.

Bei diesem Ansatz wird das Klassifizierungsproblem als mehrdimensionaler Vektor dargestellt, in dem die empirischen Daten abgebildet werden. Eine Menge dieser Vektoren bildet zusammen mit einer zugeordneten Klasse eine Trainingsbasis. Diese Trainingsbasis wird in einem überwachten Lernverfahren genutzt, um eine Funktion zu generieren, welche aus einem gegebenen Eingabevektor möglichst genau die zugehörige Klasse ermittelt.

Diese Lernverfahren ermöglichen die Erzeugung eines integrierten Schätzers, welcher die Funktion zur Relevanzschätzung direkt enthält, ohne dass die Existenz- und Signifikanzwahrscheinlichkeiten separat bestimmt werden müssen. Da ein so erzeugter Relevanzschätzer genau die Anforderungen einer einzelnen Fahrerassistenzapplikation berücksichtigt, muss für jede Fahrerassistenzapplikation ein eigener Relevanzschätzer innerhalb der Sensordatenfusion generiert werden.

Die Eigenschaften eines solchen Schätzers sollen hierbei die im Abschnitt 3.2 genannten allgemeinen Anforderungen für eine Sensordatenfusion widerspiegeln. Im Speziellen muss dieser dann folgende Eigenschaften aufweisen:

- **Genau**: Der Relevanzschätzer soll bereits nach wenigen Messzyklen relevante Objekte von irrelevanten möglichst sicher trennen können.
- **Robust**: Der Relevanzschätzer soll gegenüber Störungen, Messrauschen und Messausreißern möglichst unempfindlich sein.
- **Probabilistisch**: Der Relevanzschätzer soll eine statistisch korrekte Aussage über die Relevanzwahrscheinlichkeit liefern.
- **Echtzeitfähig**: Der Relevanzschätzer soll innerhalb einer definierten Zeitschranke eine Objekthypothese klassifizieren.
- **Modular**: Der Relevanzschätzers soll leicht auf andere Sensorik und auf andere Relevanzdefinitionen adaptierbar sein.

Um einen integrierten Schätzer zu generieren, sind verschiedene Schritte notwendig. Diese bestehen aus der Definition des Zustandsraumes, der Erzeugung von Trainingsdaten, der Anwendung von Algorithmen aus dem Bereich des „Maschinellen Lernens“, deren Verifikation und evtl. Nachbearbeitung der Schätzergebnisse. Diese einzelnen Schritte werden in den nachfolgenden Abschnitten beschrieben. Im Fokus bleiben dabei die genannten Anforderungen, sodass die Teilschritte auch daraufhin untersucht werden, ob sie diese negativ beeinträchtigen.

Um die Modularitätsanforderung zu beachten, basiert die folgende Klassifikation auf eine reine Parametrisierung, sodass der Relevanzschätzer für unterschiedliche Sensorkonfigurationen und Funktionen in unterschiedlichen Instanzen verwendet werden kann. Die Relevanzschätzer werden zum einen aus Parametern des Klassifikationsverfahrens (z. B. Gewichte für ein künstliches neuronales Netz oder Stützvektoren für eine Support Vector Machine) und zum anderen aus Parametern zur Beschreibung des Eingabevektors instanziiert: Eine Änderung der Implementation des Relevanzschätzers ist nicht notwendig.

4.6.1. Definition des Zustandsraumes

Bevor der Zustandsraum zur Relevanzbestimmung definiert wird, ist es zunächst notwendig, relevante Objekte für ein Fahrerassistenzsystem nach Definition 3 zu beschreiben. Im Folgenden ist ein Fahrerassistenzsystem

zum hochautomatischen Fahren auf Autobahnen und autobahnähnlichen Domänen exemplarisch im Fokus. Als relevant anzunehmen seien alle Objekte, die sich auf einem durchgehenden Fahrstreifen befinden. Hierzu zählen keine Objekte, die über- oder unterfahrbar sind und auch keine Objekte, die sich komplett auf Stand-, Verzögerungs- oder Beschleunigungsstreifen befinden.

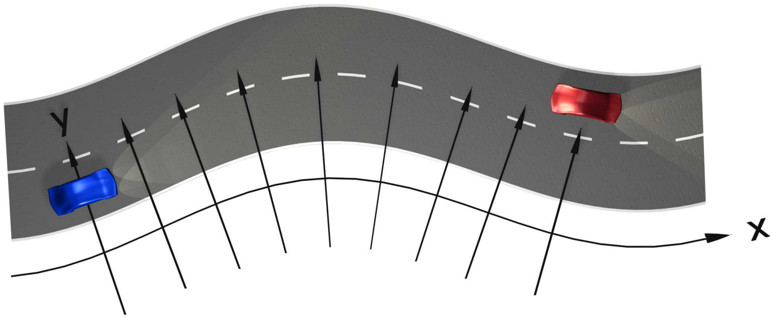


Abbildung 4.7.: Koordinatensystem der Fahrbahn

Um einen Zustandsvektor zu erzeugen, der alle wesentlichen Faktoren zur Relevanzbestimmung enthält, ist die Kombination verschiedener Teilfusionen notwendig.

Die Objekte in der Objektfusion werden in einem kartesischen Koordinatensystem gemäß [ISO 2011] verfolgt. Für die Entscheidung, ob ein Objekt signifikant sein könnte, ist entscheidend, ob es sich auf der Straße bzw. im Fahrstreifen des Ego-Fahrzeuges befindet. Hierzu findet eine Koordinatentransformation auf Basis der Fahrbahnschätzung statt, welche die Objekthypothesen in Bezug zur Fahrbahnschätzung setzt.

Das Fahrbahnkoordinatensystem zeichnet sich dadurch aus, dass dessen X-Achse dem Fahrbahnverlauf folgt und die y-Achse jeweils senkrecht auf der X-Achse steht. Es kann auch als eine Akkumulation von Fahrzeugkoordinatensystemen nach [ISO 2011] betrachtet werden, sofern das Referenzfahrzeug eine konstante Ablage zur Fahrbahn einhält. Ein solches Fahrbahnkoordinatensystem für eine Straße ist exemplarisch in Abbildung 4.7 dargestellt.

Die Transformation ist theoretisch nicht immer eindeutig: Dort wo sich die Y-Achsen schneiden, gibt es Punkte, die verschiedene X-Werte nach der Transformation erzielen können. Um eine solche Konstellation zu erreichen, sind praktisch aber Kurven mit sehr starker Krümmung erforder-

lich. Zusätzlich sind noch Sensoren notwendig, die einen so großen Erfassungsbereich haben, dass sie die Objekte im fraglichen Bereich detektieren können.

Da die untersuchte Domäne im Bereich der Autobahnen und autobahnähnlichen Straßen angesiedelt ist, können Kurven mit einer kritischen Krümmung hinsichtlich der gegebenen Sensorkonfiguration ausgeschlossen werden. Praktisch ist somit eine eindeutige Koordinatentransformation immer gegeben.

1. Gütemaß aus Objektfusion	2. Anzahl von unterschiedlichen Sensoren, die das Objekt je gemessen haben	3. Alter des Objektes
4. Objekt wurde im letzten Zyklus durch Messung bestätigt	5. Relativ vom Ego-Fahrzeug geschätzte Fahrstreifenzuordnung	6. Abstand des Objektes zur linken Fahrstreifenmarkierung
7. Abstand des Objektes zur rechten Fahrstreifenmarkierung	8. Längsabstand im Fahrbahnkoordinatensystem	9. Querabstand im Fahrbahnkoordinatensystem
10. Längsgeschwindigkeit im Fahrbahnkoordinatensystem über Grund	11. Quergeschwindigkeit im Fahrbahnkoordinatensystem über Grund	12. Längsbeschleunigung im Fahrbahnkoordinatensystem über Grund
13. Konturlänge des Objektes	14. Konturbreite des Objektes	15. Gütemaß aus Radarsensor, falls vorhanden

Tabelle 4.5.: Verwendete Attribute zur Relevanzschätzung

Nicht nur die Position, sondern auch die Längs- und Quergeschwindigkeiten sowie die Längsbeschleunigung gehen im Fahrbahnkoordinatensystem transformiert in den Zustandsvektor mit ein. Hierzu wird zunächst die Richtungsänderung der Fahrbahn zwischen Ego-Fahrzeug und Objekt betrachtet. Der Geschwindigkeits- bzw. Beschleunigungsvektor des Objektes wird dann entgegen dieser Richtungsänderung gedreht, sodass die X- und Y-Komponenten der Vektoren relativ zur Fahrbahn ausgerichtet sind.

Weitere Attribute des Zustandsvektors sind eine nach Abschnitt 4.4.5 erzeugte Existenzwahrscheinlichkeit, das zuletzt assoziierte Gütemaß aus

dem verwendeten Radarsensor[4], das Alter der Objekthypothese und weitere Attribute, welche in Tabelle 4.5 dargestellt sind. Für diesen Zustandsvektor wird im Folgenden eine Relevanzwahrscheinlichkeit bestimmt, welche direkt für das Fahrerassistenzsystem genutzt werden kann.

Bei der Auswahl der Attribute sind solche berücksichtigt worden, die die Relevanzschätzung positiv beeinflussen könnten. Ob dieses bei allen Attributen wirklich der Fall ist, wird später in Abschnitt 4.6.5 untersucht. Insgesamt ist bei diesem Auswahlschritt nur ein geringes Expertenwissen notwendig: Dieser darf dem Maschinellen Lernverfahren keine wesentliche Attribute unterschlagen, und sollte möglichst wenig irrelevante Attribute vorgeben. Eine Vorgabe, wie die einzelnen Attribute zu gewichten sind, ist nicht nötig.

4.6.2. Prospektive Erzeugung der Trainingsdaten

Die Ergebnisqualität von Algorithmen aus dem Bereich des „Maschinellen Lernens“ ist wesentlich von der Güte der Trainingsdaten abhängig. Diese sollten sowohl repräsentativ als auch umfassend sein. Aufgrund der vielen möglichen Zustände von Fahrbahn und anderen Objekten und der hohen Dimension des Zustandsraumes ist hierfür eine große Menge von Trainings- und Referenzdaten notwendig.

Um diese mit einem begrenzten Aufwand zu erzeugen, sind Verfahren notwendig, die diese zum größten Teil aus Messdaten automatisch generieren. Eine Möglichkeit besteht in der Verwendung zusätzlicher Referenzsensorik, welche bei widersprüchlichen Messdaten einen Hinweis auf die richtige Tendenz gibt. Nachteilig an dem Verfahren ist jedoch, dass man eine Referenzsensorik benötigt, die um Größenordnungen besser ist, als die verwendete. Auch der zusätzliche Aufwand durch Verbau und Auswertung darf nicht vernachlässigt werden.

Alternativ kann man das Mittel der Prospektion verwenden, um Referenzdaten zu erzeugen. Hierbei wird der Umstand genutzt, dass in aufgezeichneten Messdaten für jeden Messzeitpunkt jederzeit in die Zukunft geschaut werden kann. Mittels dieser prospektiven Analyse können praktisch alle Objekte korrekt als relevant oder als nicht relevant klassifiziert werden.

Die Prospektion erzeugt zum einen für die gesamten Messdaten eine Ground Truth-Fahrbahn auf Basis der Stützpunkte nahe dem Ego-Fahrzeug. Dort ist die Fahrbahnschätzung um ein Vielfaches genauer als

[4] Der Algorithmus zur Generierung des Gütemaßes ist unbekannt, da er vom Sensorhersteller nicht offen gelegt wurde.

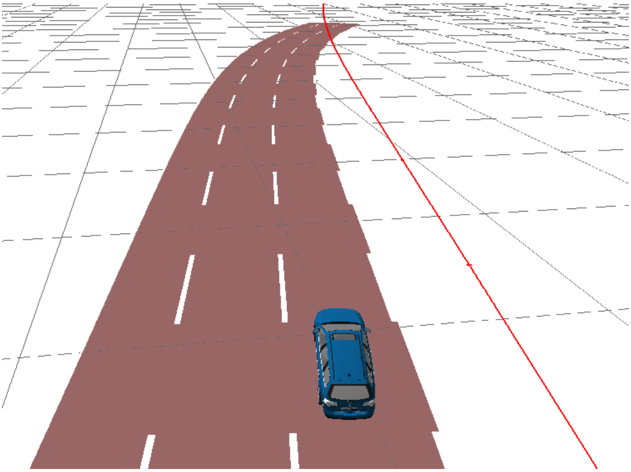


Abbildung 4.8.: Aktuelle und prospektive Fahrbahnverlaufsschätzung (Straße bzw. rote Linie). Je paralleler die Straße zur Referenzlinie verläuft, desto genauer ist die aktuelle Fahrbahnverlaufsschätzung.

die nichtprospektive Schätzung in großer Entfernung. Abbildung 4.8 zeigt eine Straßenschätzung und die prospektive Fahrbahnreferenzlinie (rote Linie). Die Ground Truth-Fahrbahn wird beispielsweise dazu verwendet, Randbebauungen nahe der Fahrbahn zu erkennen und als irrelevant zu markieren.

Weitere Regeln betreffen den prospektiven Objektzustand. So werden beispielsweise stationäre Objekthypothesen nur dann als relevant angenommen, wenn diese von mindestens zwei Messungen aus unterschiedlichen Sensoren assoziiert worden sind, sich dem Ego-Fahrzeug bis auf 15 Meter Luftlinie nähern und eine Position aufweisen, die das Objekt signifikant macht. Dynamische Objekte werden aufgrund des prospektiven Trackingalters klassifiziert. Sehr schnelle Objekte benötigen ein deutlich geringeres prospektives Alter, wenn diese das Ego-Fahrzeug mit hoher Geschwindigkeit passieren. Von Ultraschallsensoren generierte Objekthypothesen im Seitenbereich werden nur als relevant betrachtet, wenn sich auf der entsprechenden Seite ein Fahrstreifen befindet, das Objekt in diesen erheblich hineinragt und eine minimale Trackingdauer prospektiv erreicht wird.

Falls ein Objekt durch die zahlreichen Regeln trotzdem falsch klassifiziert worden ist, gibt es noch die Möglichkeit, diese Objekthypothese manuell zu kennzeichnen. Zu diesem Zweck wurden alle Daten gesichtet und mit synchron aufgezeichnetem Videomaterial oder mit einem bekannten Testaufbau verglichen und gegebenenfalls korrigiert.

Als Trainingsbasis dienen diverse Autobahnfahrten inklusive Schilderbrücken und Baustellen. Auf Autobahnen ist als kritisches Szenario ein Stauende denkbar, bei dem sich das Ego-Fahrzeug einem stationären Ziel mit hoher Geschwindigkeit nähert. Um solch ein Szenario erlernen zu können, sollte es auch in den Trainingsdaten enthalten sein. Da ein Stauende in realen Messdateien selten vorkommt, wurde es mit Prüfkörpern auf zwei verschiedenen Testgeländen nachgestellt. Hierbei wurden Einschränkungen bzgl. der Repräsentativität der Trainingsdaten in Kauf genommen.

Insgesamt sind dabei über 700.000 Objekthypothesen mit zugehöriger Relevanzklasse erzeugt worden, wobei deutlich mehr irrelevante Objekte als relevante verzeichnet worden sind: Der Anteil der relevanten Objekte betrug dort 21,6%.

4.6.3. Anforderungen und Auswahl von Maschinellen Lernalgorithmen

Die im vorherigen Abschnitt erzeugten Datensätze sollen im Folgenden genutzt werden, um einen Klassifikator zu konstruieren, der relevante Objekte von nichtrelevanten zu unterscheiden vermag. Da es beispielsweise bei Scheinzielen und stationären Zielen denkbar ist, dass diese auf gleiche oder sehr ähnliche Muster basieren, ist ein perfekter Klassifikator unrealistisch. In diesem Fall sollte dieser eine Wahrscheinlichkeit ausgeben, die dem Anteil der relevanten Objekte in der Gesamtmenge der Objekte entspricht, die ein solches Klassifikationsmuster aufweisen. Der Klassifikator soll also eine realitätsnahe probabilistische Schätzung unterstützen.

Hierzu sollte die probabilistische Ausgabe idealerweise auf einer quadratischen Minimierung der Fehlerfunktion zwischen Ein- und Ausgabemuster basieren (vgl. [Alpaydin 2004, S. 29 ff]). Hierzu sei $\kappa(x)$ die Klassifikatorfunktion, $M = \{m_1, \ldots, m_n\}$ die Menge der Muster und o_i die jeweils eindimensionale erwartete Ausgabe. Die Klassifikatorfunktion ist genau dann statistisch korrekt, wenn sie trotz einer konstanten Fehlklassifikationsrate[5] darauf basiert, folgende durchschnittliche Fehlerfunktion $e(M)$ zu minimieren:

[5] Eine Fehlklassifikation liegt im probabilistischen Fall genau dann vor, wenn $|\kappa(m_i) - o_i| > 0,5$ beträgt

$$e(M) = \frac{1}{n}\sum_{i=1}^{n}(\kappa(m_i) - o_i)^2 \tag{4.29}$$

Dass ein Ergebnis statistisch korrekt ist, wenn die Gleichung 4.29 erfüllt ist, sei an einer Menge von ähnlichen Mustern erläutert. Diese Muster liegen aufgrund ihrer Ähnlichkeit dicht im Merkmalsraum beieinander und bilden einen sog. „Cluster“. Gegeben sei ein solcher Cluster M_C, der eine Multimenge von $n = |M_C|$ nicht unterscheidbaren Klassifikationsmustern m enthält. In dieser Multimenge befinden sich t Muster, hinter denen sich relevante Objekte verbergen. Da die Klassifikationsmuster nicht unterscheidbar sind, liefert die Klassifikatorfunktion κ für jedes Element m des Clusters das gleiche Funktionsergebnis. Für dessen Fehlerfunktion gilt:

$$e(M_C) = \frac{1}{n}\left((n-t)\cdot\kappa(m)^2 + t\cdot(\kappa(m)-1)^2\right) \tag{4.30}$$

Durch Ableitung der Fehlerfunktion $e(M_C)$ kann man zeigen, dass $\kappa(m)$ Gleichung 4.29 genau dann minimiert, wenn:

$$\kappa(m) = \frac{t}{n}, \forall m \in M_C \tag{4.31}$$

Damit die statistische Korrektheit nicht nur für einen Cluster gilt, sondern für den gesamten Merkmalsraum, müssen alle Klassifikationsmuster einem Cluster zugeordnet werden. Für die Berechnung der Relevanzwahrscheinlichkeiten für die einzelnen Cluster gilt dann 4.30 entsprechend.

Liefert die Klassifikatorfunktion nur ein statistisch ungenaues Gütemaß, so lässt sich dieses dennoch durch eine Korrekturfunktion nutzen. Die Erstellung einer solchen Korrekturfunktion wird später in Abschnitt 4.6.8 beschrieben.

Neben einer gewichteten Klassifikation muss ein geeigneter Algorithmus noch weitere Kriterien erfüllen: Zum einen sollte er keine weiteren Modellannahmen benötigen, da auf eine expertenbasierte Modellannahme hier verzichtet werden soll. Solche Ansätze gehören der Kategorie der sog. „statistischen mustererkennenden Algorithmen“ an. Zum anderen sollte der Algorithmus auch bei höherdimensionalen Klassifikationsproblemen echtzeitfähig sein. Dieses Kriterium betrifft nur den Klassifikationsteil des Algorithmus, nicht aber dessen Trainingsphase.

Im Folgenden sollen verschiedene Klassifikationsalgorithmen bewertet werden, ob diese potenziell als Relevanzklassifikator dienen könnten. Eine Auswahl wird später genauer untersucht und mit Referenzdaten untereinander verglichen. Bekannte probabilistische Klassifikationsalgorithmen sind unter anderen:

- Bayessches Netz (s. Anhang A.3.1)
- k-Nearest-Neighbour (s. Anhang A.3.2)
- Lineare Support Vector Machine (s. Anhang A.3.3)
- Support Vector Machine mit Kernelerweiterung (s. Anhang A.3.4)
- Künstliches neuronales Netz (s. Anhang A.3.5)

Zwar sind die genannten Machine-Learning-Verfahren in der Lage, mittels Lerndaten gewichtete Klassifikatorfunktionen bereitzustellen. Einige davon scheinen jedoch die speziellen Anforderungen, die ein Relevanzklassifikator mit sich bringt, weniger gut umzusetzen als andere:

Eine wichtige Anforderung für solch einen Klassifikator ist seine Echtzeitfähigkeit,welche insbesondere für den kNN-Algorithmus schwierig umzusetzen ist. Prinzipbedingt muss dieser auf eine im Vorfeld unbekannte Teilmenge der Trainingsmenge in Echtzeit zugreifen können. Je nach Speichertechnologie, Suchalgorithmus und Menge der Trainingsdaten kann dieses Kriterium schwierig zu erfüllen sein. Da später eine sehr große Menge an Trainingsdaten verwendet wird, ist eine mangelnde Echtzeitfähigkeit zu erwarten, weshalb auf die weitere Untersuchung dieses Klassifikators verzichtet wird.

Als weitere Anforderung galt es, dass dieser möglichst genaue gestaltet werden soll. Problematisch können dabei expertenbasierte Modellannahmen und -vereinfachungen sein, da hierbei das komplexe Problem der Objektrelevanz genau verstanden werden muss. Ist dieses nicht fehlerfrei möglich, so können falsche Randbedingungen einen negativen Einfluss auf die Genauigkeit des Schätzers ausüben.

Das Bayessche Netz erfordert jedoch die Definition der Netztopologie und der Diskretisierung kontinuierlicher Attribute. Wie dargestellt, gibt es zwar verschiedene Algorithmen, welche die einzelnen Aspekte schätzen können, falls kein gutes expertenbasiertes Modell generiert werden kann. Problematisch ist jedoch die kombinatorische Komplexität, die ohne Randbedingungen zu bewältigen ist: So sind die Erlernung einer guten Diskretisierung, einer guten Netztopologie und die daraus ermittelten Verbundwahrscheinlichkeiten der einzelnen Knoten nicht unabhängig. Damit ist ein insgesamt multiplikativer Aufwand bei einer großen Trainingsbasis festzustellen, um diese Optimierungsalgorithmen zu kombinieren.

Da dem Bayesschen Netz außerdem bei vielen Klassifizierungsproblemen eine unterdurchschnittliche Genauigkeit im Vergleich zu vielen ande-

ren Algorithmen nachgesagt wird (vgl. [Kotsiantis 2007][6] und [Ling 2002][7]), wird auf die weitere Untersuchung dieses Klassifikators verzichtet.

Ein einfaches Verfahren stellt die lineare Trennung dar. Dieses hat den großen Vorteil, dass es die Lernphase sehr schnell absolviert, dabei die Trainingsdaten stark kondensiert, keine weiteren Parameter wie Abstandsmaße benötigt und auch sehr schnell Daten klassifiziert. Die resultierende Ebenengleichung ist vergleichsweise leicht zu interpretieren, wodurch das Verfahren transparenter als andere ist. Nachteilig ist die zu erwartende geringere Klassifikationsgüte bedingt durch die deutlich geringere Anzahl an freien Parametern. Falls das Problem der Relevanzklassifikation jedoch so einfach ist, dass dieser Algorithmus ein gleichwertiges Klassifikationsergebnis hervorbringt, ist es den anderen Verfahren aufgrund der vielen positiven Eigenschaften überlegen. Aus diesem Grunde wird die lineare Trennung weiter untersucht.

	kNN	**BN**	**Lineare Trennung**	**SVM**	**ANN**
Genauigkeit	o	o	-	++	+
Klassifikationsgeschwindigkeit	-	++	++	++	++
Transparenz	- -	o	+	-	-
Modellannahmen	o	-	++	+	+

Tabelle 4.6.: Bewertung verschiedener Algorithmen aus dem Bereich des „Maschinellen Lernens“ (vgl. [Kotsiantis 2007], [Caruana 2006])

Die Support Vector Machine und das künstliche neuronale Netz hingegen sind komplexere Machine-Learning-Verfahren, die auch in der Lage sind, nichtlineare Klassifikationsprobleme gut zu approximieren. Beiden Verfahren wird zudem eine hohe Klassifizierungsgeschwindigkeit nachgesagt, sodass diese das größte Potenzial haben, Genauigkeit und Echtzeitfähigkeit zu vereinen. Nachteilig ist, dass beide Verfahren an vielen Stellen durch diverse Parameter modifiziert werden können. So stehen beispielsweise verschiedene Kernelfunktionen für eine Support Vector Machine zur

[6] In [Kotsiantis 2007] wird gezeigt, dass Daten mit kontinuierlichen Attributen, hohen Abhängigkeiten und redundanten Daten deutlich schwieriger durch Bayessche Netze klassifiziert werden können als mit anderen Verfahren.

[7] In [Ling 2002] wird bewiesen, dass für einfache n-XOR-Funktionen bzw. Paritätsfunktionen ein Bayessches Netz mit hoher Komplexität benötigt wird.

Auswahl, während beim künstlichen neuronalen Netz die Anzahl der versteckten Schichten und deren Neuronenzahl als Parameter festgelegt werden müssen. Hier muss ein Experte eine günstige Parametrisierung finden. Im Unterschied zum Bayesschen Netz haben sich aber einige wenige Standardparametrisierungen unabhängig vom speziellen Problem herauskristallisiert, welche bei vielen Klassifikationsproblemen gute Ergebnisse liefern. Diese werden in den folgenden Abschnitten verwendet, um diese beiden Verfahren zu untersuchen.

Eine Übersicht über die Vor- und Nachteile der einzelnen Verfahren ist in Tabelle 4.6 zusammengefasst.

4.6.4. Anwendung der Lernverfahren zur Relevanzklassikation

Die im Abschnitt 4.6.2 erzeugte Lernbasis wird in Trainings-, Referenz- und Testdaten getrennt und normiert, sodass sich jedes Attribut im Intervall [0, 1] befindet. Diese Daten werden anschließend in verschiedenen Lernverfahren genutzt, die die Klassifizierungsfunktion approximieren. Insgesamt sind die drei im vorherigen Abschnitt ausgewählten Verfahren getestet worden: Lineare Trennung, künstliche neuronale Netze und Support Vector Machine.

Wie erwartet, zeichnet sich die lineare Trennung während des Lernverfahrens durch eine sehr hohe Geschwindigkeit aus. Diese beträgt nur einen Bruchteil der anderen Lernverfahren.

Einen recht hohen Trainingsaufwand verzeichnete das künstliche neuronale Netz. Hierbei wurden verschiedene Parametrisierungen getestet, wobei sich eine einzelne versteckte Schicht mit 30 Neuronen als geeignet herausstellte. Bei der Verwendung von deutlich mehr oder weniger Neuronen zeigte das künstliche neuronale Netz ein schlechteres Klassifikationsergebnis bei den Testdaten, was durch Überanpassung bzw. Unteranpassung erklärt werden kann. Es wurden auch verschiedene Aktivierungsfunktionen getestet: Hierbei lieferte die logistische sigmoide Aktivierungsfunktion das beste Ergebnis auf den Testdaten.

Als Ergebnis sind die Netzstruktur und die erlernten Verbindungsgewichte in Abbildung 4.9 dargestellt. Je stärker das Gewicht zwischen zwei Neuronen, desto dicker ist die Verbindungslinie zwischen diesen dargestellt. Positive Gewichte (blaue Verbindungen) erhöhen die elektrische Aktivität bei den verbundenen Neuronen und erhöhen damit auch insgesamt tendenziell die Netzaktivität. Umgekehrt dämpfen negative Gewichte (rote Verbindungen) bei entsprechender Eingabe diese.

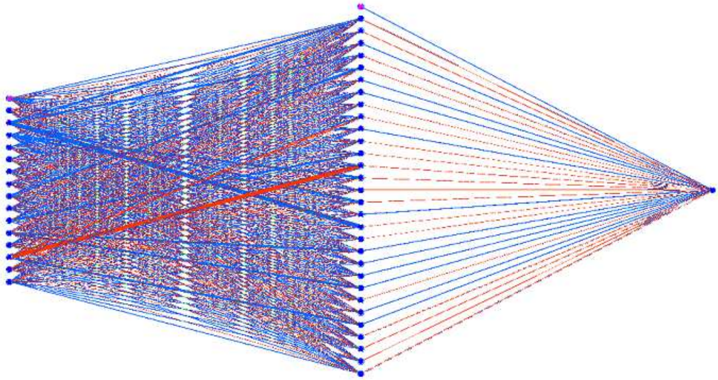


Abbildung 4.9.: Eingesetztes künstliches neuronales Netz mit 48 Neuronen. Blau: Positive Gewichte, Rot: Negative Gewichte.

Den höchsten Trainingsaufwand verzeichnete die Support Vector Machine. Hierbei fällt der Nachteil auf, dass die Anzahl der verwendeten Supportvektoren nicht begrenzt ist und diese tendenziell mit zunehmender Lernmenge ansteigt. Da die Lernmenge mit ca. 300.000 Datensätzen sehr groß ist, und die Support Vector Machine potenziell eine sehr große Anzahl an freien Parametern aufweist, kommt es hier möglicherweise zur Überanpassung: Wie die Übersicht über die einzelnen Verfahren in Tabelle 4.7 zeigt, sind während der Lernphase über 20.000 Supportvektoren mit je 15 Dimensionen entstanden. Hierbei wurden auch verschiedene C-Konstanten getestet (vgl. Gleichung A.17), jedoch lieferten diese sehr ähnliche Ergebnisse. Bei den Kernelfunktionen hat die radiale Kernelfunktion die beste Klassifikationsleistung erbracht.

Als Ergebnis bzgl. Genauigkeit liefert das künstliche neuronale Netz das beste Ergebnis mit 95,7% auf die Referenzmenge. Ein mittleres Ergebnis liefert die Support Vector Machine gefolgt von der linearen Trennung.

4.6.5. Dimensionsreduktion des Zustandsraumes

Bei der Erzeugung des Zustandsraumes in Abschnitt 4.6.1 sind solche Attribute ausgewählt worden, die potenziell den Zustandsraum so erweitern, dass eine Trennung durch funktionsapproximierende Algorithmen erleichtert wird. Unklar ist bisher, ob dafür alle Attribute mit der Objektrelevanz

Maschinelles Lernverfahren	Lernparametertyp	Anzahl gelernter Parameter	Genauigkeit bzgl. Referenzmenge
Lineare Trennung	Koeffizienten der Ebenengleichung	15	93,1%
Support Vector Machine	Support-Vektoren	23935 · 15	94,2%
Künstliches neuronales Netz	Verbindungsgewichte	496	95,7%

Tabelle 4.7.: Freie Parameter verschiedener Algorithmen aus dem Bereich des Maschinellen Lernens

korrelieren bzw. ob diese redundant vorliegen. So ist beispielsweise anzunehmen, dass die Trackingdauer eines Objektes im hohen Maße mit der klassischen Existenzwahrscheinlichkeitsberechnung korreliert, ohne dass man direkt sagen könnte, welches der beiden Attribute das wichtigere ist.

Eine unnötige Vergrößerung des Zustandsraumes zieht mehrere Nachteile mit sich. Zum einen benötigen Lern- und Klassifikationsalgorithmen mehr Rechenzeit, sodass sich der Aufwand vergrößert, diese zu nutzen. Ein kleinerer Zustandsraum kann somit auch die Echtzeitfähigkeit verbessern. Schwerer wiegt zum anderen aber die potenziell schlechtere Performance: Je hochdimensionaler der Zustandsraum ist, desto mehr Lerndaten braucht man für seine Abdeckung. Sind Bereiche nur mit sehr geringer Dichte vorhanden, so können sich durch partielle Überanpassungen Scheinzusammenhänge ausprägen.

Aus diesem Grund wird im Folgenden versucht, unwichtige Attribute zu identifizieren und zu eliminieren. Wie im vorherigen Abschnitt gezeigt, liefert das künstliche neuronale Netz in diesem Fall das beste Klassifikationsergebnis. Es bildet somit im Folgenden die Referenz für die Aussage, ob ein Attribut wichtig oder unwichtig ist.

Aus dem mehrdimensionalen Zustandsraum wird sukzessiv ein Attribut entfernt und das Netz neu angelernt. Liefert es auf einer Referenzmenge ein ähnlich gutes oder sogar ein besseres Klassifikationsergebnis, so erscheint das getestete Attribut als unwichtig. Da die Ausprägung des künstlichen neuronalen Netzes nicht nur von den Lerndaten, sondern auch von einer zufälligen Gewichtsinitialisierung abhängt, können sich unterschiedliche Netze bei gleichen Lerndaten ausbilden. Da sie abhängig von der

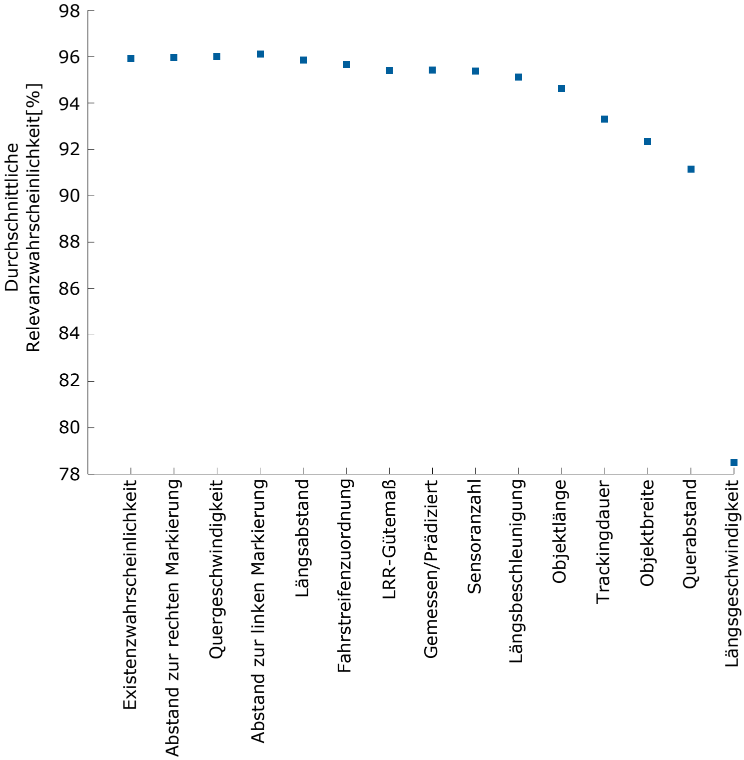


Abbildung 4.10.: Durchschnittliche Klassifikationsrate nach Entfernung einzelner Attribute

Initialisierung während des Lernprozesses unterschiedliche lokale Minima und somit eine unterschiedliche Performance erreichen könnten, erscheint es vorteilhaft, mehrere Netze mit unterschiedlichen Initialisierungen anzulernen und die Ergebnisse zu mitteln.

Bei der Entfernung eines von mehreren möglichen Attributen wird dasjenige ausgewählt, welches nach Entfernung aus dem Zustandsraum im Mittel über alle Lerndurchläufe die höchste Korrektklassifikationsrate liefert. Dieses hat den geringsten positiven Einfluss auf das Netz. Somit vermeidet man bei dieser Methode einen Einfluss auf die Testreihenfolge der Attribute.

Der Zustandsraum wird im Folgenden solange verkleinert, bis man kein

Attribut mehr findet, welches man entfernen kann, ohne dass sich die Korrektklassifikationsrate verschlechtert. Es zeigt sich, dass es nur wenige Attribute gibt, bei deren Entfernung sich eine bemerkbare Verschlechterung des Klassifikationsergebnisses ergibt. Hierzu zählen die Längsgeschwindigkeit und der Querabstand. Die anderen Merkmale scheinen nur einen geringen Beitrag zur korrekten Klassifikation beizusteuern. Abbildung 4.10 zeigt die Veränderung der Klassifikationsrate nach sukzessiver Entfernung einzelner Attribute.

Aus dieser Abbildung ist auch zu entnehmen, dass der Zustandsraum bis auf 11 Dimensionen reduziert werden kann. Der Abstand zur linken und rechten Markierung, die nach konventionellen Methoden berechnete Existenzwahrscheinlichkeit und die Quergeschwindigkeit des Objektes sind danach unwichtige bzw. kompensierbare Attribute. Der Klassifikator erreicht hierbei eine durchschnittliche Klassifikationsrate von 96,1%.

Die wichtigsten Attribute sind nach dieser Analyse die Längsgeschwindigkeit, der Querabstand, Objektbreite und Trackingdauer. Während die Längsgeschwindigkeit stationäre Ziele von dynamischen separiert und somit ein wichtiges Kriterium für die Erkennung existierender Objekte ist, trennt der Querabstand vermutlich nicht signifikante von signifikanten Objekten. Die Trackingdauer kann kurzfristige Scheinziele unterdrücken und kompensiert somit wohl die Existenzwahrscheinlichkeit nach konventionellen Methoden.

Anhand der Objektbreite hat das Netz indirekt einen Anhaltspunkt, von welchen Sensoren der Objekttrack aktualisiert worden ist. Der Radarsensor und der Ultraschallsensor können keine Objektkontur schätzen, sodass hier bestimmte Vorgabewerte genutzt werden. Die Stereokamera kann hingegen eine Breite bzw. Länge messen, sodass die Objekthypothese meist eine Konturschätzung ungleich der Vorgabewerte aufweist.

Das gezeigte Verfahren liefert insgesamt allerdings nur ein lokales Minimum bei der Bestimmung der optimalen Attributmenge. Die Suche nach diesem hat nur eine polynomielle Komplexität zur Attributanzahl n von $O(n) = n^2$. Um das globale Minimum zu finden, müssten hingegen alle Netze mit allen möglichen Attributkombinationen getestet werden. Dieses Verfahren weist eine exponentielle Komplexität von $O(n) = 2^n$ auf und ist somit bei einer größeren Anzahl von Attributen zu aufwendig. Diese Grenze ist bereits im vorliegenden Fall überschritten, sodass die Suche nach dem lokalen Minimum als Optimierungsmethode mit geringerer Komplexität gewählt worden ist.

4.6.6. Tiefpassfilterung des Ergebnisses

Der zu klassifizierende Zustandsvektor enthält zum einen Attribute, die kumulativ während des Trackings erzeugt werden. Hierzu zählen beispielsweise die Trackingdauer oder auch die Anzahl von unterschiedlichen Sensoren, die das Objekt je zuvor gemessen haben.

Die meisten anderen Attribute bilden jedoch nur eine Momentaufnahme der Schätzung, wobei aktuelle Schätzungen der Objektfusion und der Fahrbahnfusion einfließen, ohne dass frühere Ergebnisse berücksichtigt werden. Insbesondere bei weit entfernten Objekten kann dies dazu führen, dass bei schlechter Fahrbahnschätzung diese Objekte mal mehr oder weniger mit einem Fahrstreifen assoziiert werden. Dadurch kann sich im Verlauf der Relevanzschätzung eine hohe Varianz ergeben. Dieses ist vor allem bei konstanten Schwellen problematisch, da das Fahrerassistenzsystem dann das Objekt alternierend als relevant und nicht relevant betrachtet.

Das Einbringen alter Relevanzschätzungen in Lernalgorithmen ist nicht ohne Weiteres möglich, da diese zum Lernzeitpunkt nicht feststehen. Außerdem verändern sie sich mit jedem Lernschritt, sodass sie theoretisch nach jeder Iteration neu erzeugt werden müssen. Es ist zusätzlich fraglich, ob die einzelnen Verfahren bei sich ständig verändernden Lerndaten konvergieren.

Aus diesem Grund wurde ein einfaches Tiefpassfilter erster Ordnung hinter der Relevanzschätzung implementiert, welcher die Relevanzwerte mittelt. Sei r_n die vorherige gefilterte Relevanzschätzung, p die ungefilterte neue Relevanzschätzung und $F \in [0,1]$ ein Faktor, der die Stärke des Tiefpassfilters angibt. Dann wird der neue gefilterte Relevanzwert r_{n+1} wie folgt berechnet:

$$r_{n+1} = F \cdot r_n + p \cdot (1 - F) \tag{4.32}$$

Vorteilig ist eine robustere und zum Teil auch genauere Relevanzschätzung, welche insgesamt konsistentere Ergebnisse liefert. Nachteilig ist eine je nach Stärke des Filters mehr oder weniger verzögerte Detektion relevanter Objekte: Erstmalig initialisierte Objekte haben häufig eine geringe Relevanzwahrscheinlichkeit. Wenn sich mit zunehmenden Objekthypothesenalter aus Sicht des Machine Learning Algorithmus herausstellt, dass diese doch relevant sind, verhindert das Tiefpassfilter eine schnelle Adaption an die aktuellen Schätzungen. Dieses ist insbesondere bei stationären Objekten nachteilig, da diese für das Fahrerassistenzsystem erst später relevant werden.

Auch kann eine gegebene statistische Korrektheit eines Schätzers durch die Tiefpassfilterung beeinträchtigt werden. So sind durch den oben genannten Effekt in vielen Fällen zu niedrige Relevanzwerte zu erwarten.

Trotz dieser theoretischen Nachteile ergab die praktische Anwendung des Tiefpassfilters auf den Relevanzklassifikator eine Verbesserung der Klassifikationsergebnisse.

4.6.7. Auswertung mittels Grenzwertoptimierungskurven

Ein häufig genutztes Verfahren, um einen probabilistischen Klassifikator zu beurteilen, ist die Erstellung einer Kurve zu dessen Operationscharakteristik. Die sog. „Grenzwertoptimierungskurve“ (engl. Receiver Operating Characteristic (ROC) - Kurve) stellt dabei die Klassifikationsrate der korrekt positiven Elemente über der Klassifikationsrate der falsch positiven Elemente dar. Für eine Einführung zu Grenzwertoptimierungskurven sei auf [Fawcett 2004] verwiesen.

Zur Bewertung eines Klassifikators wird häufig dessen diskretes Flächenintegral unter seiner Grenzwertoptimierungskurve als Gütemaß benutzt. Diese beträgt bei einem Zufallsklassifikator 0,5 und bei einem perfekten Klassifikator 1.

	Lineare Trennung	**SVM**	**ANN**
Gütemaß (Flächenintegral)	0,960	0,969	0,972
Standardfehler des Flächenintegrals	0,00060	0,00057	0,00097
Konfidenzintervall für 95%	0,959 - 0,961	0,967 - 0,970	0,971 - 0,973

Tabelle 4.8.: Kenngrößen der Klassifikatoren nach einer Grenzwertoptimierungsanalyse

Im Folgenden wird jeweils eine Grenzwertoptimierungskurve für die drei erstellten Klassifikatoren berechnet und die Ergebnisse in Tabelle 4.8 dargestellt. Es zeigt sich, dass die Lineare Trennung das schlechteste Ergebnis und das künstliche neuronale Netz das beste Ergebnis liefert. Das Gütemaß bescheinigt der Support Vector Machine hingegen ein mittleres Ergebnis. Da sehr viele Ground Truth-Daten für die Kurvenerstellung genutzt

werden konnten, ist die Standardabweichung des diskreten Flächenintegrals sehr gering. Bei wenigen Ground Truth-Daten können größere Diskretisierungsfehler auftreten, welche sich in der Standardabweichung niederschlagen.

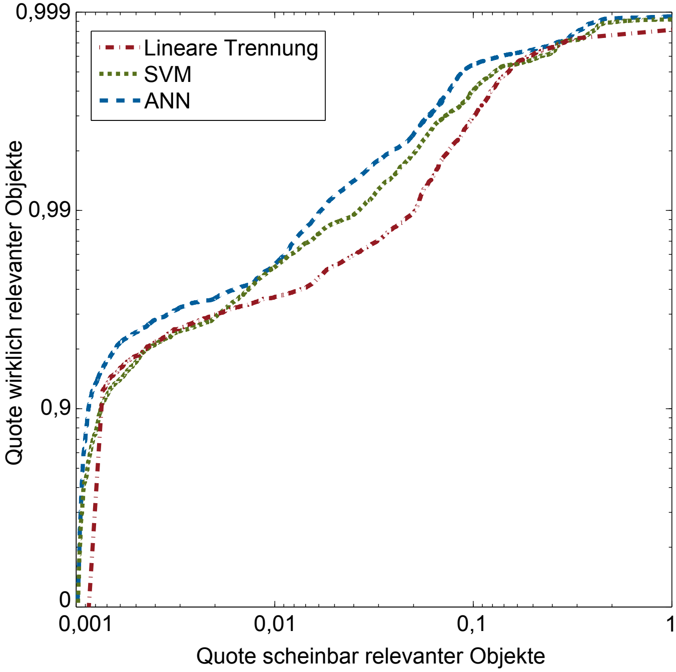


Abbildung 4.11.: Doppellogarithmische Darstellung von Grenzwertoptimierungskurven für verschiedene Klassifikatoren

Insgesamt liefern alle Verfahren so gute Korrektklassifikationsraten, sodass für die Visualisierung der Grenzwertoptimierungskurve in Abbildung 4.11 eine doppellogarithmische Darstellung gewählt worden ist, um die Unterschiede besser herauszuheben. Durch diese Darstellung fehlen bestimmte Bereiche: So haben alle Kurven $(0/0)$ als Startpunkt und $(1/1)$ als Endpunkt, welche hier fehlen.

Aus dieser Abbildung kann man entnehmen, dass das künstliche neuronale Netz in allen wesentlichen Bereichen stets besser als die Support Vector Machine und die lineare Trennung ist. Die Lineare Trennung hinge-

gen ist bei einigen wenigen Bereichen geringfügig besser als die Support Vector Machine.

4.6.8. Histogrammanalyse und -korrektur

Einige Lernverfahren optimieren nativ ihre Zielfunktion durch eine quadratische Minimierung des Schätzfehlers. Wie im Abschnitt 4.6.3 gezeigt, ist eine solche Zielfunktion dann bereits statistisch korrekt, da diese genau durch eine statistisch korrekte Relevanzwahrscheinlichkeit minimiert wird. Im Anhang A.3.5 wird gezeigt, dass das künstliche neuronale Netz auf solch einer Zielfunktion basiert, im Allgemeinen jedoch nicht die Support Vector Machine weder mit noch ohne Kernelfunktion.

Trotzdem können Verfahren, die ihre Zielfunktion quadratisch minimieren, durch ungünstige Wahl von Lerndaten oder falsche Parametrisierung diese Minimierung nur auf Basis von Lerndaten ausführen. Bei neuen Testdaten könnten diese schlechtere und evtl. statistisch inkorrekte Ergebnisse liefern. Auch die im Abschnitt 4.6.6 vorgestellte Tiefpassfilterung kann negativen Einfluss auf die statistische Korrektheit haben.

Aus diesem Grund ist eine Überprüfung auf statistische Korrektheit und ggf. die Erstellung einer Korrekturfunktion vorteilhaft. Im Folgenden wird dazu nach [DeGroot 1982][8] ein Histogramm mittels Ground Truth-Daten generiert, welches die Basis solch einer Korrekturfunktion bildet.

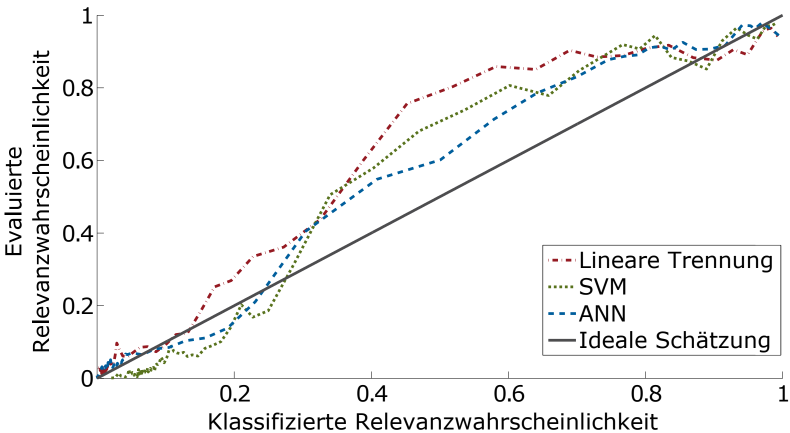


Abbildung 4.12.: Korrekturfunktionen basierend auf dem Histogramm

[8]Vergleiche hierzu den Abschnitt A.1.2 im Anhang.

Die Histogrammerstellung erfolgt folgendermaßen: Von einer großen Menge von neuen Trainingsdaten wird die Relevanzwahrscheinlichkeit bestimmt. Nun werden diese in gleich große Mengen unterteilt, sodass sich in jeder Teilmenge die Datensätze befinden, die eine möglichst ähnliche Relevanzwahrscheinlichkeit haben. Die Relevanzwahrscheinlichkeitsschätzung von Datensätzen einer Teilmenge ist genau dann statistisch korrekt, wenn diese äquivalent zum Quotienten von relevanten Datensätzen und der Gesamtmenge an Datensätzen ist. Ein Schätzer ist immer dann statistisch korrekt, wenn das Schätzergebnis auf dieses Verhältnis abgebildet wird. Die Abbildungsfunktionen für die lineare Trennung, die Support Vector Machine und das künstliche neuronale Netz sind in Abbildung 4.12 dargestellt. Die geschätzte Relevanzwahrscheinlichkeit nach dem Tiefpassschritt wird auf die evaluierte Relevanzwahrscheinlichkeit abgebildet.

Mithilfe der Histogrammanalyse können die Ergebnisse jeder beliebige Klassifikationsfunktion statistisch korrekt abgebildet werden, sodass man die Relevanzwerte direkt verwenden kann. So würde eine Zufallsfunktion nach Histogrammanalyse und -korrektur beispielsweise jeden Relevanzwert auf einen konstanten Wert abbilden, da in jeder Teilmenge das Verhältnis zwischen relevanten und irrelevanten Objekten gleich groß ist.

Wichtig bei der Erstellung ist außerdem eine abgestimmte Wahl der Klassengröße. Ist diese zu klein, kommt es zur Überanpassung. Ist diese zu groß, wird der Wertebereich, welcher die Teilmenge repräsentiert unnötig vergrößert. Dies hat zur Folge, dass bestimmte Relevanzwerte innerhalb dieses Wertebereiches schlechter diskriminiert werden.

Um die Diskriminierung naher Relevanzwerte zu verbessern, wird eine lineare Interpolation zwischen den Klassen benachbarter Wertebereiche durchgeführt. In Abbildung 4.13 ist die Methode dargestellt: Die lineare Interpolation (rote Linie) der Relevanzwahrscheinlichkeit $p(x)$ erfolgt zwischen zwei benachbarten Klassen (gestrichelte Rechtecke), die jeweils einen Relevanzbereich repräsentieren. Die Höhe der Rechtecke basiert auf dem Verhältnis von relevanten Objekten (blaue Quadrate) zur Gesamtzahl der Objekte dieser Klasse. Diese lineare Interpolation nimmt an, dass die Relevanzfunktion eine gewisse Stetigkeit aufweist.

Angewandt auf die drei Klassifizierungsfunktionen lineare Trennung, Support Vector Machine und künstliches neuronales Netz ergeben sich die in Abbildung 4.12 dargestellten Korrekturkurven. Bei diesen ist insbesondere zu sehen, dass alle drei Schätzfunktionen keine sicheren relevanten Ereignisse schätzen können: Die Schätzwerte „0“ und „1“ werden nur auf nichtsichere Werte nahe „0“ bzw. „1“ korrigiert. Auch sind die Relevanzschätzungen in weiten Teilen des Wertebereiches zu niedrig, sodass

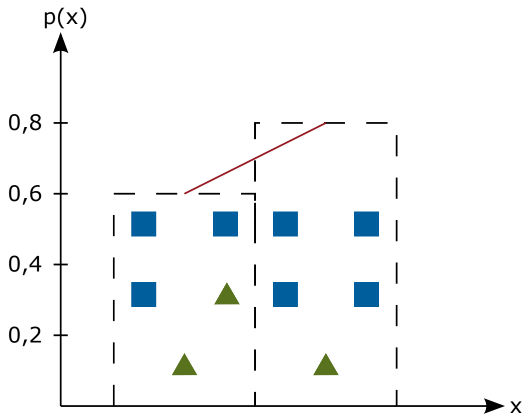


Abbildung 4.13.: Lineare Interpolation zwischen diskreten Klassen innerhalb eines Histogramms

die Korrekturkurven meist über der Kurve eines idealen Schätzers liegen. Dessen Korrekturfunktion entspricht dabei der identischen Abbildung. Zu sehen ist auch, dass die Abweichung der Korrekturkurve des künstlichen neuronalen Netzes zur identischen Abbildung am geringsten ist. Daraus folgt, dass dieser Schätzer noch am ehesten statistisch genaue Resultate liefert.

Wie optimistisch ein Schätzer insgesamt ist, lässt sich aus einer mittleren Abweichung $\varsigma(R)$ berechnen. Diese ergibt sich aus dem arithmetischen Mittel der Abweichungen von den beiden sicheren Schätzwerten „0“ und „1“ und kann aus einer Menge von Relevanzschätzungen $R = \{r_1, \ldots, r_n\}$ wie folgt berechnet werden:

$$\delta(r_i) = \begin{cases} 1 - r_i & \text{wenn } r_i > 0,5 \\ r_i & \text{sonst} \end{cases} \tag{4.33}$$

$$\varsigma(R) = \frac{1}{n} \sum_{i=1}^{n} \delta(r_i) \tag{4.34}$$

Je optimistischer ein Schätzer ist, desto näher liegt $\varsigma(R)$ bei null. Je schlechter die Korrektklassifikationsrate eines Schätzers ist, desto pessimistischer sollte dieser sein und dieses mit einer hohen mittleren Abweichung zu den idealen Werten benennen.

	Mittlere Abweichung	**Korrektklassifikationsrate**	**Falsch Positive**	**Falsch Negative**
Lineare Trennung	0,066	93,9%	2,8%	3,3%
SVM	0,103	95,0%	2,6%	2,4%
ANN	0,049	96,3%	1,7%	2,0%

Tabelle 4.9.: Schätzergebnisse und Standardabweichung

Bei der Betrachtung der Ergebnisse in Tabelle 4.9 zeigt sich, dass die Support Vector Machine wesentlich pessimistischere Schätzungen als die Lineare Trennung erzeugt, obwohl diese bessere Klassifikationsergebnisse liefert. Das künstliche neuronale Netz postuliert entsprechend seiner guten Korrektklassifikationsrate die geringste mittlere Abweichung zu den idealen Schätzwerten.

Die Korrektklassifikationsraten haben sich bei allen Schätzern durch die Tiefpassfilterung und Histogrammkorrektur gleichermaßen um zwei Promille verbessert. Dies bedeutet eine Verringerung der Fehlalarmrate um ca. 5 %.

4.6.9. Bewertung

Durch die Histogrammkorrektur wird jeder Schätzer statistisch korrekt, sodass dieser Punkt nicht mehr untersucht werden muss. Insgesamt verbleiben dann noch drei Kriterien zur Bewertung: Das Wichtigste ist die erreichte Genauigkeit eines Schätzers. Die Auswertung kann beispielsweise durch dessen Grenzwertoptimierungskurve oder dessen Korrektklassifikationsrate erfolgen. Letztere wurde bereits in Tabelle 4.9 dokumentiert.

Als bester Schätzer hat sich in jeder Zwischenstufe das künstliche neuronale Netz herausgestellt: Im Originalzustand, nach dessen Dimensionsreduktion als auch nach der Tiefpassfilterung und Histogrammkorrektur liefert dieser jeweils das genaueste Ergebnis.

Verschiedene beispielhafte Klassifikationen des künstlichen neuronalen Netzes sind in Abbildung 4.14 dargestellt. Mehrere Objekthypothesen (weiße Kästen) werden durch Daten des Radar- (rote Sterne) und Stereokamerasensors (grüne Kästen) gebildet. Diese werden anschließend mit der aus Fahrbahnmarkierungen (schwarze Linienzüge) und Leitplankenschätzung (grüne Linien) erzeugten Straßenschätzung verknüpft. Fahrzeuge werden in die weißen Kästen eingeblendet, wenn das künstliche neuronale Netz die zugehörigen Objekthypothesen als relevant klassifiziert hat. In diesem



Abbildung 4.14.: Darstellung klassifizierter Objekthypothesen (weiße Boxen) aus dem Objekttracking. Fahrzeuge werden in die Objekthypothesen eingeblendet, wenn deren Relevanzwahrscheinlichkeit mehr als 50 % beträgt.

Beispiel werden die beiden vorausfahrenden Fahrzeuge richtigerweise als relevant klassifiziert, während Leitplanken, Gegenverkehr und ein statisches Scheinziel unter der Brücke korrekt als irrelevant betrachtet werden.

Algorithmus	**Genauigkeit der Relevanzschätzung**	**Echtzeitfähigkeit**	**Transparenz**
Lineare Trennung	-	++ (<0,1 ms)	+
SVM	o	- (≈220 ms)	- -
ANN	+	+ (<1,0 ms)	-

Tabelle 4.10.: Bewertung der Verfahren zur Relevanzklassifikation

Auch wenn der künstliche neuronale Netz-Klassifikator das beste Ergebnis liefert, ist dieses bei Weitem nicht perfekt. Grund dafür ist eine im Abschnitt 4.3 sehr weitreichende Definition für relevante Objekte. Diese wurde so gewählt, um den Klassifikatoren eine möglichst anspruchsvolle Datenbasis zu liefern. Es sind beispielsweise keine Entfernungs- oder Dynamikbeschränkungen gemacht worden, sodass insbesondere Messungen schwer zu klassifizieren sind, die neue Objekthypothesen weit entfernt initialisieren. Im Markt befindliche Fahrerassistenzsysteme würden die Signifikanz deutlich enger auslegen, sodass existente Objekte länger im nicht-

signifikanten Bereich beobachtet werden können. Würde man für solche Fahrerassistenzsysteme abgestimmte starke Randbedingungen integrieren, so würden sich die Klassifikationsergebnisse noch wesentlich verbessern.

Da die Relevanzschätzung bei bestimmten Fahrerassistenzsystemen ein Bestandteil der Längsregelung ist, ist eine geringe Totzeit für dessen Regelungsqualität vorteilhaft. Aus diesem Grund müssen alle Schätzer bestimmte Echtzeitkriterien erfüllen. So müssen sie innerhalb sehr kurzer Zeit für eine größere Objektmenge deren jeweilige Relevanz bestimmt haben[9]. Dieses Kriterium erfüllt natürlich die lineare Trennung, aber auch das künstliche neuronale Netz kann bei einer optimierten Implementation diese Anforderung bewältigen. Bei der Support Vector Machine hingegen ist aufgrund der vielen Support-Vektoren eine deutliche Latenz messbar, was dieses Verfahren in dieser Form ungeeignet macht.

Wünschenswert wäre auch eine größere Transparenz der Verfahren. So bilden sowohl Support Vector Machine als auch das künstliche neuronale Netz eine Art „Black-Box“, die auf vielen Parametern basiert. Beim künstlichen neuronalen Netz lassen sich noch wichtige Verbindungspfade herausfinden, hingegen ist bei der Support Vector Machine aufgrund der vielen Support-Vektoren keine Transparenz mehr vorhanden.

Eine Zusammenfassung der Vor- und Nachteile der einzelnen Verfahren ist in Tabelle 4.10 beschrieben.

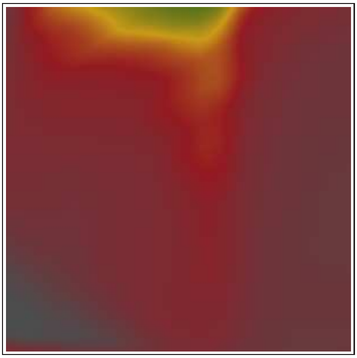
4.7. Ableitung von Regeln aus einem integrierten Relevanzschätzer

Der vorgestellte integrierte Schätzer bildet eine hochdimensionale Wahrscheinlichkeitsdichtefunktion, die je nach Machine-Learning-Algorithmus mehr oder weniger intransparent ist. Um diese zu verstehen, werden im Folgenden Schnitte durch den hochdimensionalen Raum durchgeführt und visualisiert.

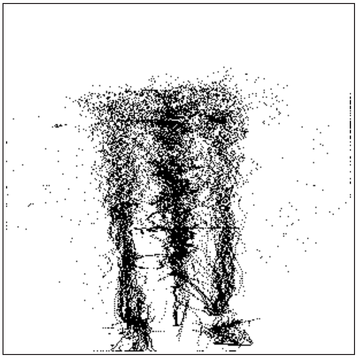
Die nicht visualisierten Dimensionen sollen dabei vernachlässigt werden. Exemplarisch wird der künstliche neuronale Netz-Klassifikator dargestellt; das Verfahren ist jedoch auf beliebig andere probabilistische Klassifikatoren anwendbar.

Während die Support Vector Machine unbekannte Attribute unterstützt, ist dieses bei der linearen Trennung und beim künstlichen neuronalen

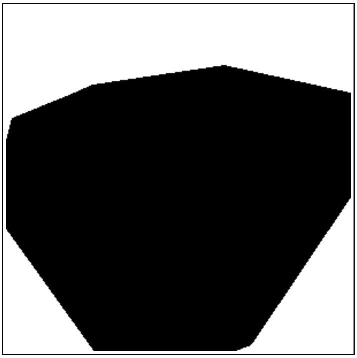
[9] Als Test wurden 32 Tracks auf einem CompactPCI-Computer ausgerüstet mit einem bis zu 2,2 GHz getakteten x86-Doppelkernprozessor und 4 GB Arbeitsspeicher klassifiziert. Die Ergebnisse des Tests sind in Tabelle 4.10 zusammengefasst.



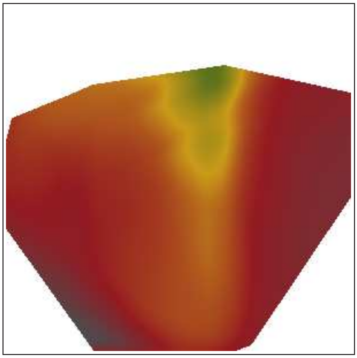
(a) Schnittbild zur Objektrelevanz bei Variation der Position



(b) Reales Auftreten der Objektposition in Messungen



(c) Minimales konvexes Polygon in der alle auftretenden Messungen enthalten sind



(d) Anwendung des Polygons auf das Schnittbild und Neuerstellung der Farbskala

Abbildung 4.15.: Entstehung von eingegrenzten Schnittbildern zur Relevanzklassifikatorvalidierung. In diesem Beispiel wird die Relevanzwahrscheinlichkeit in Abhängigkeit der Objektposition visualisiert.

Netz nicht der Fall. Daher wird die Menge der Testdaten genutzt, um den Einfluss der anderen Attribute zu neutralisieren. Um einen Bildpunkt im Schnittbild zu erzeugen, werden die zu visualisierenden Attribute ent-

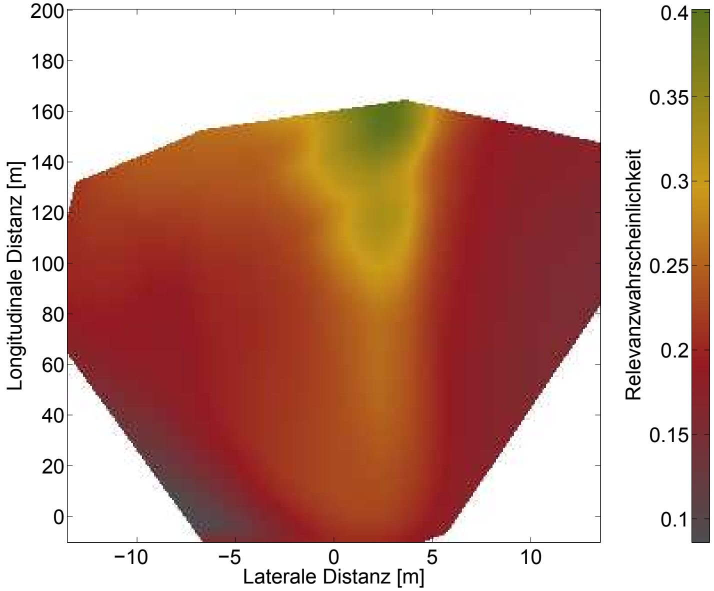


Abbildung 4.16.: Durchschnittliche Relevanz bei variabler fahrstreifenbezogener Objektposition

sprechend der Bildkoordinate in der gesamten Datenmenge gesetzt. Anschließend wird diese Menge durch ein künstliches neuronales Netz klassifiziert. Die durchschnittliche Relevanzwahrscheinlichkeit entspricht dann dem arithmetischen Mittel der Klassifikationsergebnisse. Für jeden einzelnen Bildpunkt muss das künstliche neuronale Netz somit ca. 200.000 unterschiedliche Testdaten durchlaufen.

Beim systematischen Setzen von mehreren Attributen tritt das Problem auf, dass diese nicht zwingend in der Lernmenge repräsentiert sind. Dadurch können Bereiche auftreten, die aus der Extrapolation weit entfernter Lernpunkte stammen. Diese Bereiche wurden im Lernprozess durch Daten nicht gestützt, sodass der Klassifikator hier beliebige Funktionsergebnisse liefern kann, ohne dass dieses während des überwachten Lernens bestraft wird. Aus diesem Grund wird zumindest das Auftreten der zu visualisierenden Attribute berücksichtigt: Hierzu wird ein minimales konvexes Polygon aus den Messwerten gebildet, welches als Maske für das finale Schnittbild dient.

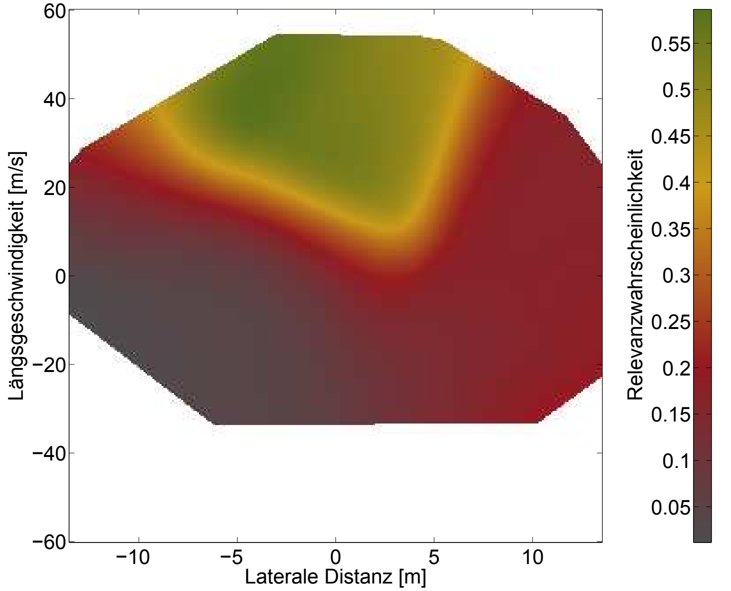


Abbildung 4.17.: Durchschnittliche Relevanz bei variabler fahrstreifenbezogener Längsgeschwindigkeit und lateralem Abstand

Abbildung 4.15 zeigt den Entstehungsprozess exemplarisch anhand der Objektposition bestehend aus fahrbahnbezogenem Längs- und Querabstand. Das Schnittbild (Abb. 4.15a) zeigt im Bereich weit entfernter Objekte in Längsrichtung einen stark interpolierten Bereich (grüner Bereich), welcher nicht durch reale Messungen (Abb. 4.15b) gestützt wird. Anhand dieser Messungen wird ein minimales konvexes Polygon (Abb. 4.15c) erstellt, welches in etwa den Sensorsichtbereich repräsentiert. Hierbei fallen Objekte weg, die sich weit links und rechts und auf der Höhe des Ego-Fahrzeugs befinden oder eine große Entfernung aufweisen. Die Anwendung dieser Maske erfordert eine Neuberechnung der Farbskala, sodass im finalen Schnittbild (Abb. 4.15d) gut die Relevanzunterschiede real auftretender Bereiche dargestellt werden.

Problematisch bleibt jedoch, dass möglicherweise Kombinationen nichtvisualisierter Attribute erzeugt werden, die so nicht in den Messdaten auftreten würden: Beispielsweise sei hier die Fahrstreifenzuordnung in Kombination von Längs- und Querabstand genannt. Sind letztere Attribute sehr

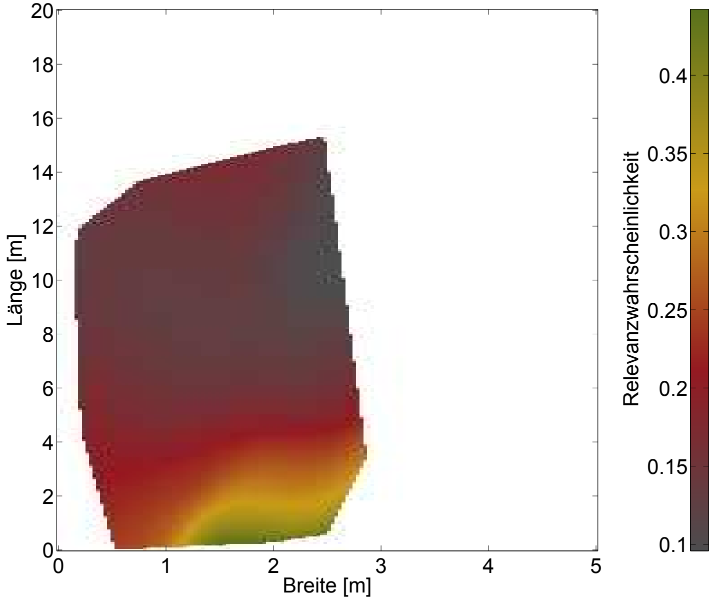


Abbildung 4.18.: Durchschnittliche Relevanz bei variabler Objektgröße

klein, so wird sich das Objekt auf dem Ego-Fahrstreifen mit definiertem Index befinden. Das Setzen eines hohen Querabstandes bei Belassung des Fahrstreifenindexes würde eine unrealistische Attributkombination darstellen, welche kaum korrekt bewertet werden kann. Eine Neutralisierung dieses Effektes ist jedoch schwierig, sodass die folgenden Schnittbilder eine eventuell ungenaue Relevanzskala aufweisen. Sie sind jedoch gut geeignet, Tendenzen darzustellen.

Ein eingegrenztes Schnittbild mit solch einer Relevanzskala ist in Abbildung 4.16 visualisiert. Es zeigt sich wie erwartet, dass Objekte mit hohem Querabstand tendenziell weniger relevant sind. Bemerkenswert ist, dass weit entfernte Objekte relevanter sind, als Objekte mit mittlerer oder niedriger Entfernung. Eine mögliche Erklärung ist der überdurchschnittlich hohe Radarquerschnitt von existenten Objekten. Diese können in höherer Entfernung als Scheinziele mit häufig niedrigem Radarquerschnitt detektiert werden.

Einen hohen Einfluss auf die Objektrelevanz hat die Längsgeschwindigkeit: In Abbildung 4.17 ist diese zusammen mit dem Querabstand darge-

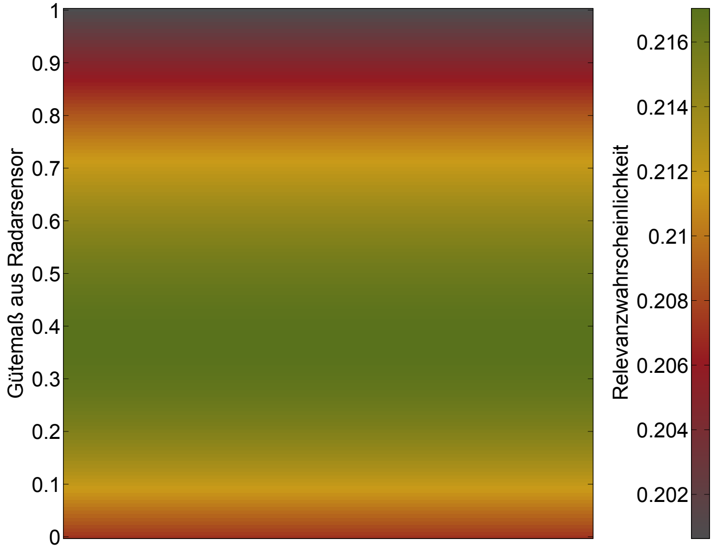


Abbildung 4.19.: Durchschnittliche Relevanz bei variablen Gütemaß aus der letzten Messung des Radarsensors

stellt. Je höher diese ist, desto relevanter ist das Objekt. Hierbei dominiert es weitestgehend den Querabstand. Dieser hat im Vergleich zur Längsgeschwindigkeit mäßigen Einfluss auf die Schätzung. Es zeigt sich, dass Objekte mit negativer Geschwindigkeit wenig Relevanz besitzen. Dies ist damit erklärt, dass als Domäne Autobahnen und autobahnähnliche Straßen gewählt worden sind, wo kein Gegenverkehr auf der eigenen Fahrbahn in den Lerndaten auftrat. Objekte mit negativer Geschwindigkeit sind somit der Nachbarfahrbahn zuzuordnen und als irrelevant zu klassifizieren.

Abbildung 4.18 zeigt den Einfluss der Objektlänge und -breite. Objekte mit einer Breite außerhalb der „normalen“ Fahrzeugbreite sind tendenziell irrelevant. Für die Länge gilt dieses nur für sehr lange Objekte. Somit werden beispielsweise Leitplankenobjekte mit großer Länge und kleiner Breite als nichtrelevant klassifiziert. Bemerkenswert ist, dass Objekte mit sehr kurzer Länge von weniger 0,5 Meter einen positiven Einfluss auf die Objektrelevanz haben. Dies ist damit erklärt, dass der Kamerasensor die Länge eines Objektes nur messen kann, wenn er dieses von der Seite her beobachten kann. Dies ist häufig nicht der Fall, wenn man einem relevanten Fahrzeug auf dem gleichen Fahrstreifen folgt. In diesem Fall nimmt die

Objektfusion nur eine sehr kurze Länge des Fahrzeuges an.

Abbildung 4.19 zeigt den alleinigen Einfluss des Radarsensorgütemaßes. Obwohl dieses bei praktisch allen Objekten verfügbar ist, hat es nur einen minimalen Einfluss auf die Objektrelevanz, was der geringe Skalenunterschied zeigt. Es kann somit durch andere Parameter aus Objekt- und Fahrbahnfusion größtenteils kompensiert werden. Wenig erwartet wurde, dass Objekte mit hohem Gütemaß tendenziell eine geringere Objektrelevanz besitzen als Objekte mit mittlerem Gütemaß. Eine mögliche Erklärung ist, dass der Radarsensor bei vielen existenten stationären Randobjekten direkt nach der ersten Messung ein hohes Gütemaß liefert. Diese Objekte werden aber auch nach kurzer Zeit direkt verworfen, da sie den Sichtbereich schnell verlassen. Relevante dynamische Objekte hingegen können ein mittleres Gütemaß erreichen, indem sie den Sichtbereich langsam verlassen. Hier prädiziert der Radarsensor intern weiter und verringert im Tracking das Gütemaß.

Insgesamt können die Schnittbilder bei dem Verständnis der generierten Wahrscheinlichkeitsdichtefunktion helfen. Sie können zusätzlich einen Experten dabei unterstützen, statische Regeln aus diesen zu generieren. Diese sind transparenter als die hochdimensionale Wahrscheinlichkeitsdichtefunktion und lassen sich auch auf Plattformen mit geringer Speicher- und Rechenleistung portieren.

4.8. Zusammenfassung

Dieses Kapitel hat sich damit beschäftigt, eine statistisch genaue Wahrscheinlichkeit zu berechnen, ob ein Objekt von einem Fahrerassistenzsystem berücksichtigt werden muss oder nicht. Hierzu wurden diverse klassische Verfahren zur Existenz- und Signifikanzschätzung vorgestellt und bewertet. Diese zeichnen sich durch viele Modellvereinfachungen und -annahmen aus und berücksichtigen teilweise nur wenige Attribute.

Im Gegensatz dazu wurde das Konzept eines integrierten Relevanzschätzers vorgestellt. Dieser erstellt aus Lerndaten ein implizites Umgebungsmodell, wobei komplexe lineare und nichtlineare Zusammenhänge erfasst und integriert werden. Dadurch kann dieser Schätzer eine Genauigkeit erreichen, wie sie nur schwer durch manuell festgelegte Regeln übertroffen werden kann.

Seine Schätzgrundlage basiert nicht nur auf den Messdaten eines einzelnen Umfeldsensors, sondern auf einer komplexen Sensordatenfusion für eine technologisch heterogene Sensorkonfiguration. Hierbei fließt unter anderem eine Straßenschätzung mit in die Bewertung ein, womit detektier-

te Fahrstreifenmarkierungen, digitale Karten und andere Faktoren berücksichtigt werden können. Im Gegensatz dazu verwenden klassische Verfahren nur eine geringe Anzahl von Attributen und machen viele Annahmen zu Abhängigkeiten, Unabhängigkeiten und Verteilungen. Auch bei der Vereinigung von Existenz- und Signifikanzwahrscheinlichkeit werden bei diesen Verfahren ähnliche Annahmen getroffen.

Der vorgestellte Schätzer hingegen integriert beide Teilwahrscheinlichkeiten zur Relevanzwahrscheinlichkeit und berücksichtigt viele Attribute. Eine Erweiterung um neue Attribute ist leicht möglich. Viel Aufwand wurde betrieben, um den probabilistischen Schätzer auch statistisch genau zu gestalten. Somit liefert er eine echte Wahrscheinlichkeit und kein Gütemaß mit unbekannter Skala.

Das Verfahren zeichnet sich ferner dadurch aus, dass der Kern des Klassifikationsalgorithmus austauschbar ist. Bei den untersuchten Kernalgorithmen erscheint das künstliche neuronale Netz als geeignet: Es vereinigt eine gute Klassifizierungsrate mit einer geringen Berechnungslatenz. Nachteilig ist eine verminderte Transparenz des Verfahrens. Ein sehr schnelles und transparenteres Verfahren stellt die lineare Trennung dar, welche allerdings eine geringere Genauigkeit aufweist.

Die Klassifikatoren wurden anhand einer sehr weitreichenden Definition relevanter Objekte getestet. So wurde von ihnen auch gefordert, erstmalig initialisierte Objekte in großer Entfernung korrekt zu klassifizieren. Unter diesen schwierigen Bedingungen erreichen diese eine Korrektklassifikationsrate von bis zu 96,3%, wobei der Klassifikationsschwellwert bei 50% lag. Für die meisten Fahrerassistenzapplikationen ist diese Rate noch zu niedrig, sodass der Schwellwert entsprechend angehoben werden muss. Dies führt meist zu einer verzögerten Relevanzklassifikation, da Objekthypothesen tendenziell eine längere Beobachtungsdauer aufweisen müssen, bevor diese hohe Schwellwerte überschreiten. Neben Objekthypothesen mit sehr kurzen Beobachtungszeiten sind auch stationäre Objekte in hoher Entfernung schwer zu klassifizieren: Diese weisen meist eine Relevanzwahrscheinlichkeit auf, welche keinen sehr hohen Schwellwert überschreitet und bleiben somit in den ersten Messzyklen meist unberücksichtigt.

Diese Eigenschaften sind jedoch nicht von Nachteil, wenn man sie mit konventionellen Objektklassifikatoren vergleicht. Diese nutzen starke Nebenbedingungen, welche einen vergleichbaren Effekt erzielen: So wird häufig eine minimale Trackingdauer bzw. eine minimale Anzahl an Messbestätigungen vorausgesetzt oder es werden stationäre Objekte im Vorfeld herausgefiltert (vgl. [ISO 2002, 2008]). Dies vereinfacht den Zustandsraum erheblich und erlaubt somit eine deutliche Steigerung der Klassifikationsrate, da schwer zu klassifizierende Objekthypothesen aus dem Zustands-

raum ganz einfach herausdefiniert worden sind.

Fraglich bei diesen konventionellen Ansätzen ist jedoch, ob wirklich alle Objekthypothesen irrelevant sind, die unter solche Nebenbedingungen fallen: So kann beispielsweise ein vor Kurzem verdecktes Objekt sofort für ein ACC-System relevant werden, auch wenn es noch keine minimale Trackingdauer erreicht hat. Es scheint außerdem von Kundennutzen, wenn ein ACC-System auf alle relevanten stationären Objekte verzögern würde und dabei keine Einschränkungen bzgl. Dynamik und Entfernung macht.

Der vorgestellte Relevanzklassifikator hingegen versucht auf solche expliziten Nebenbedingungen zu verzichten und nutzt das Gesamtbild der Objekthypothese für eine genaue Relevanzschätzung. Bei hohen Schwellwerten können zwar ähnliche Nebenbedingungen implizit entstehen, jedoch hat die maschinell angelernte und nichtlineare Trennfunktionen das Potential, deutlich komplexere Zustandsräume zu trennen, sodass die Nebenbedingungen weniger restriktiv ausfallen. Der Relevanzklassifikator kann somit einen Teil der relevanten Hypothesen, die normalerweise unter Nebenbedingungen wegfallen würden, der Fahrerassistenzapplikation zur Verfügung stellen.

Insgesamt hat die dynamische Bestimmung von Relevanzwahrscheinlichkeiten mittels Maschinellen Lernverfahren das Potenzial, Fehlauslösungs- und fehlerhafte Nichtauslösungsraten zu verringern und die nutzbare Reaktionszeit zu erhöhen.

Kapitel 5.

Probabilistische Belegungsgitterfusion

5.1. Einleitung

In den vorherigen Kapiteln wurde eine komplexe Objektfusion mit integriertem Relevanzschätzer vorgestellt, welche exemplarisch in der strukturierten Umgebung einer Autobahn bzw. autobahnähnlichen Straße ihre Domäne findet. Sie beinhaltet viele Modellannahmen, wie beispielsweise das Klothoidenmodell der Fahrbahn, ein einfaches Dynamikmodell basierend auf beschleunigten Körpern für zu verfolgende Objekte, das Boxmodell als deren Konturmodell und diverse Rauschmodelle in den einzelnen Kalmanfilterschritten. Diese Annahmen bilden Randbedingungen in den einzelnen Modellen, wodurch die Anzahl freier Variablen im Umfeldmodell potentiell reduziert wird. In den vorherigen Kapiteln wurde eine komplexe Objektfusion mit integriertem Relevanzschätzer vorgestellt, welche exemplarisch in der strukturierten Umgebung einer Autobahn bzw. autobahnähnlichen Straße ihre Domäne findet. Sie beinhaltet viele Modellannahmen, wie beispielsweise das Klothoidenmodell der Fahrbahn, ein einfaches Dynamikmodell basierend auf beschleunigten Körpern für zu verfolgende Objekte, das Boxmodell als deren Konturmodell und diverse Rauschmodelle in den einzelnen Kalmanfilterschritten. Diese Annahmen bilden Randbedingungen in den einzelnen Modellen, wodurch die Anzahl freier Variablen im Umfeldmodell potentiell reduziert wird.

Die Messdaten werden mithilfe dieser Modelle in ein strukturiertes Umfeldmodell überführt, wobei das Umfeldmodell aber letztendlich nur eine Abstraktion der Wirklichkeit darstellt. Je ungenauer die Modelle die Umgebung und Sensoreigenschaften umsetzen, desto weniger genau bildet das Umfeldmodell die Realität ab. Die Modellierung der Umgebung kann in komplex strukturierten Domänen sehr aufwendig sein, wobei die Umfeldmodelle dann sehr viele freie Parameter aufweisen, die alle korrekt geschätzt werden müssen.

Als alternativer Ansatz wurde bereits in den 80er Jahren im Robotikbereich das sog. „Belegungsgitter“ entwickelt (vgl. [Moravec 1985]), wel-

ches eine modellarme und hochdimensionale Umgebungsrepräsentation darstellt. Dieses ist für komplex strukturierte Umgebungen geeignet, wie sie beispielsweise im urbanen Bereich auftreten.

Belegungsgitter diskretisieren den Raum um den Roboter bzw. um das Fahrzeug in einzelne Zellen. Den einzelnen Messungen werden die örtlich zugehörigen Zellen zugeordnet und deren Zustand wird aktualisiert. Als Ergebnis erhält man eine Menge von Zellen mit unterschiedlichen Zellzuständen, die eine Umgebungsschätzung darstellen.

Typischerweise repräsentiert das Umfeldmodell eine statische Umgebung. Dynamische Objekte werden entweder als Störgröße behandelt oder sie bleiben bereits im Vorfeld unberücksichtigt. Einige neuere Ansätze sind aber auch in der Lage, dynamische Objekte zu behandeln (vgl. [Bouzouraa 2010]). Diese beinhalten jedoch wieder Modellannahmen, meist auch Dynamikmodelle ähnlich der Objektfusion.

Belegungsgitter wurden bereits vielfach im Robotik- und Automobilbereich erfolgreich eingesetzt. So spielte bei vielen Teilnehmern der DARPA Urban Challenge (vgl. [Effertz 2009] und [Thrun 2006]) diese Form der Umgebungsrepräsentation eine entscheidende Rolle für die Funktionalität der autonomen Fahrzeuge. Im Robotikbereich wurden diese insbesondere innerhalb von Gebäuden und begrenzten Geländearealen eingesetzt (vgl. [Elfes 1989b] und [Ferguson 2006]).

Bei praktisch allen Arbeiten steht die Befahrbarkeitsanalyse von Bereichen im Vordergrund. Die Zellen des Belegungsgitters enthalten zu diesem Zweck einen Zustand bestehend aus einem oder mehreren Attributen, der Informationen über die Befahrbarkeit der jeweiligen Zelle bereitstellt.

Um diesen Zellzustand zu schätzen, werden aus den Messungen zwei Messdatentypen generiert. Der eine Typ modelliert Belegtbereiche, welches beispielsweise Objektinformationen sein können, der andere Typ repräsentiert Freibereiche. Diese werden genutzt, um die mit den Messungen assoziierten Zellen des Belegungsgitters zu aktualisieren und eine lokale Befahrbarkeitsschätzung zu erzeugen.

Insgesamt beschäftigt sich dieses Kapitel mit der Untersuchung und Entwicklung von Algorithmen für solche Belegungsgitter, die bestimmte Anforderungen erfüllen müssen. Als Ergebnis soll die Applikation einen Bereich des Belegungsgitters in Echtzeit abfragen können und eine möglichst probabilistische, genaue und statistisch korrekte Schätzung über dessen Befahrbarkeit erhalten[1].

Wie bei der Relevanzschätzung in Kapitel 4 besteht der Nutzen einer

[1] Die Begriffe „Echtzeitfähigkeit“ und „statistische Korrektheit“ werden im Anhang A.1.1 bzw. Anhang A.1.2 definiert.

solchen probabilistischen Schätzung zum einen in der Verringerung von Fehlauslösungs- und fehlerhaften Nichtauslösungsraten und zum anderen in der Möglichkeit mittels Schwellwerten das Verhältnis dieser beiden Raten genau bestimmen zu können.

5.2. Anforderungen an eine Belegungsgitterfusion

In Kapitel 3 sind verschiedene Anforderungen für eine Sensordatenfusion entwickelt worden, die sich insgesamt auf die Softwarearchitektur und alle Teilfusionen erstrecken muss. Für eine Belegungsgitterfusion abgeleitet, sind folgende Kriterien wünschenswert:

- **Genau, Probabilistisch**: Die Belegungsgitterfusion soll eine genaue Befahrbarkeitsschätzung für eine beliebige Menge von Zellen liefern. Das Verfahren sollte innerhalb der Diskretisierungsgrenzen invariant gegenüber Zellgrößenänderungen sein.

- **Robust**: Einzelne Messausreißer und systematische Fehlziele einzelner Sensoren sollten möglichst kompensiert werden.

- **Echtzeitfähig**: Die Belegungsgitterfusion soll auf einer bestimmten Zielplattform echtzeitfähig sein.

- **Modular**: Es sollten verschiedene Sensoren und Anwendungen durch verallgemeinerte Schnittstellen unterstützt werden. Außerdem sollten verschiedene Logiken eingesetzt werden können.

Um diese Anforderungen zu bewältigen, ist zunächst ein Belegungsgitter als solches notwendig. Dazu werden zunächst im folgenden Abschnitt 5.3 grundlegende Algorithmen und Datenstrukturen für Belegungsgitter vorgestellt, welche aus der Computergrafik entnommen und adaptiert worden sind. Mit diesen ist die Verwaltung und der Zugriff auf definierte Zellmengen effizient möglich.

Im anschließenden Abschnitt 5.4 werden drei verschiedene Logikalgorithmen vorgestellt, welche die Fusionierung der Zellzustände mit Messevidenzen zu neuen Zellzuständen ermöglichen. Eine einheitliche mathematische Form dieser Algorithmen unterstützt im Sinne der Modularitätsanforderung die Austauschbarkeit.

Die schwierigste Anforderung ist die der genauen Umfeldschätzung, wofür im Abschnitt 5.5 verschiedene Ansätze aus der Literatur untersucht werden. Hierbei werden neben den Fusionsalgorithmen auch die inversen

Sensormodelle vorgestellt, welche aus Messungen Messevidenzen für Zellen generieren. Anschließend wird untersucht, welche Stärken und Schwächen der gesamte Ansatz bezüglich der genauen und statistisch korrekten Umfeldschätzung hat, wobei die Bewertung sowohl die jeweilige Anwendung berücksichtigt als auch eine mögliche Übertragung in den Automotivebereich prüft.

Bei der Untersuchung der Ansätze aus der Literatur zeigt sich, dass diese bestimmte Abhängigkeiten zwischen Zellen und Messungen nicht oder nur unzureichend auflösen. In Abschnitt 5.6 werden diese beiden Kernprobleme konventioneller Algorithmen identifiziert und formal definiert. Sie zu lösen ist der Schlüssel zur Erfüllung der Genauigkeits- und Robustheitsanforderung.

Die Lösung dieser beiden Probleme kann jeweils unabhängig betrachtet werden. In den darauf folgenden beiden Abschnitten 5.7 und 5.8 werden entsprechende Algorithmen vorgestellt, die diese Probleme berücksichtigen und somit eine genauere Schätzung als konventionelle Belegungsgitteralgorithmen erreichen. Im Sinne der Modularitätsanforderung unterstützen diese Algorithmen bis zu drei verschiedene Messtypen und eine allgemeine Polygonabfrageschnittstelle.

Nach der theoretischen Entwicklung dieser Algorithmen wird in Abschnitt 5.9 deren Latenz anhand experimenteller Ergebnisse auf einer Zielplattform ermittelt, was eine Voraussetzung für die Bewertung der Echtzeitfähigkeit darstellt. Zum Schluss wird dieses Kapitel im Abschnitt 5.10 zusammengefasst.

5.3. Stand der Technik: Algorithmen und Datenstrukturen für Belegungsgitter

Belegungsgitter repräsentieren das Umfeld in Form von Zellen, die jeweils einen Zustand beinhalten. Da die speichertechnische Realisierung eines unendlich großen Umfeldes nicht möglich ist, wird nur ein Teil des Umfeldes modelliert. Typischerweise wird nur das direkte Umfeld ortsfest modelliert, welches direkt entscheidungsrelevant für eine Fahrerassistenzfunktion bzw. Bahnplanung des Roboters ist.

Da sich der Roboter oder das Ego-Fahrzeug im Allgemeinen bewegt, gilt es eine Strategie zu entwerfen, Zellen zu verwerfen, welche irrelevant geworden sind. Ein weitverbreiteter Ansatz ist die Nutzung sog. Ringpufferspeicher, welcher hier vorgestellt wird. Dieser ermöglicht es, einen konstant großen Umgebungsausschnitt mit einer definierten Speichermenge darzustellen.

Bevor die Aktualisierung von Zellen durch eine Messung möglich ist, müssen in einem ersten Schritt zunächst die relevanten Zellen ermittelt werden. Hier sind ressourcenarme und echtzeitfähige Algorithmen gefragt, die die entsprechende Menge der Zellen und deren Koordinaten liefern. Je nach Größe und Auflösung des Belegungsgitters und Größe des assoziierten Bereiches einer Messung können sehr viele Zellen durch eine einzige Messung betroffen sein. Als Beispiel sei ein großer Freibereich genannt, der indirekt aus einer Laserscannermessung generiert wurde.

Im Folgenden werden Algorithmen vorgestellt, die Zellzugriffe in Punkt-, Linien oder Polygonform ermöglichen. Diese sind aus der Computergrafik bekannt und wurden auf die Anforderungen eines Belegungsgitters adaptiert. Neben Algorithmen zur Sensordatenfusion ermöglichen diese Zellzugriffsmethoden auch das Abfragen von definierten Flächen. Diese Funktion ist grundlegend, um beispielsweise die Befahrbarkeit eines Bereiches zu ermitteln.

5.3.1. Zweidimensionaler Ringpuffer

Eine häufig genutzte Datenstruktur für die Belegungsgitterrepräsentation ist der sog. zweidimensionale Ringpufferspeicher. Er basiert auf einem zweidimensionalen Feld, wobei jedes Element genau den Zustand einer Zelle repräsentiert. Das Ego-Fahrzeug befindet sich approximativ in der Mitte dieses ortsfesten Feldes. Bewegt sich das Ego-Fahrzeug, so werden die Zellen, welche den relevanten Bereich verlassen, als neue Ressourcen wiederverwendet.

Hierzu existieren verschiedene Strategien: In [Effertz 2009, S. 106 ff] werden Gruppen von Zellen zu Sektoren über verkettete Listen zusammengefasst und diese als Ganzes entfernt bzw. hinzugefügt. Ein anderer Ansatz nach [Kelly 1998] gruppiert die Zellmengen zu Zeilen und Spalten und verschiebt diese als Ganzes. Vorteil des sektorbasierten Ansatzes ist ein effizienteres Speicherlayout bzgl. Zellzugriffsalgorithmen innerhalb eines Sektors. Nachteilig ist die Notwendigkeit zusätzlicher Zeigerdereferenzierungschritte innerhalb der übergeordneten Datenstruktur, bevor man auf die eigentlichen Zellen zugreifen kann. Auch macht der sektorbasierte Ansatz es notwendig, für eine garantierte Größe mehr Zellen und somit mehr Speicher zu reservieren, da die Verschiebung nicht unmittelbar erfolgt, sondern erst dann, wenn eine Sektorgrenze überschritten wurde. Im Folgenden wird daher der Ansatz nach [Kelly 1998] weiter verfolgt.

Der zweidimensionale Ringpufferspeicher nutzt im klassischen Ansatz den Modulo-Operator, um eine Zelle innerhalb der Datenstruktur zu lokali-

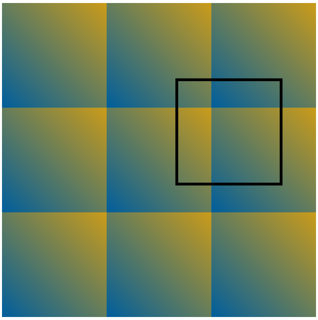


Abbildung 5.1.: Modulo-Operation bildet direktes Umfeld (schwarzes Quadrat) auf 4 disjunkte Ringpufferbereiche ab

sieren: Sei $P_{Welt} = (i_{Welt}, j_{Welt})$ der Zellindex der Zelle in der Welt und $n \times n$ die Größe des Ringpufferspeichers. Dann befindet sich die gesuchte Zelle bei $P_{Puffer} = (i_{Puffer}, j_{Puffer})$ innerhalb des Ringpufferspeichers mit:

$$i_{Puffer} = i_{Welt} \mod n \tag{5.1}$$

$$j_{Puffer} = j_{Welt} \mod n \tag{5.2}$$

Hierbei ist $\mod$ der ganzzahlige Modulo-Operator. Diese Abbildungsoperation verursacht bis zu vier Speichergrenzen im direkten Umfeld, welche in Abbildung 5.1 zwischen den farbigen Kacheln auftreten. Zu beachten ist, dass jeder beliebige Weltzellenindex auf eine Zelle im Ringpufferspeicher innerhalb des schwarzen Quadrats abgebildet wird, auch wenn dieser sich außerhalb dieses Ausschnittes befindet. Sind solche Zellzugriffe nicht auszuschließen, so müssen diese vorher mittels Bereichsprüfung abgefangen werden.

5.3.2. Punktalgorithmus

Die Bestimmung einer Zelle zu einer gegebenen Koordinate ist einerseits zur Modellierung von varianzlosen punktförmigen Messungen notwendig, andererseits aber auch grundlegend für spätere Zellzugriffsalgorithmen, die komplexere Formen aufweisen.

Zur Bestimmung einer Zelle, die zu einer gegebenen Weltkoordinate korrespondiert, ist im Wesentlichen eine Diskretisierung des Raumes erforderlich. Die Diskretisierungsgröße entspricht dabei der Zellgröße. Die

Abbildung der ortsfesten Koordinate auf die entsprechenden Weltindizes erfolgt durch eine entsprechende Skalierung.

Sei $P = (x,y)$ die ortsfeste Koordinate der abzufragenden Zelle innerhalb eines $s \times s$ großen Belegungsgitters, welches $n \times n$ Zellen enthält. Dann hat die Zelle die Weltindizes $I = (i_{Welt}, j_{Welt})$ mit:

$$i_{Welt} = \left\lfloor \frac{nx}{s} \right\rfloor \tag{5.3}$$

$$j_{Welt} = \left\lfloor \frac{ny}{s} \right\rfloor \tag{5.4}$$

Mittels der Modulo-Operation im vorangegangenem Abschnitt kann diese dann auf eine Zelle im Ringpufferspeicher abgebildet werden.

5.3.3. Linienalgorithmus

Der Zellzugriffsalgorithmus für Linien ermöglicht die Bestimmung der Menge von Zellen, welche je nach Interpretation ganz oder wesentlich von einer definierten Linie geschnitten werden.

Praktisch alle Ansätze basieren auf dem sog. Bresenhamalgorithmus, welcher ausschließlich mit Ganzzahloperationen effizient eine Linie von einem Startpunkt zu einem Endpunkt zeichnet (vgl. [Bresenham 1965]). Dieser Algorithmus bestimmt zunächst, ob die X- oder Y-Richtung die Hauptrichtung der Linie ist:

$$d_x = x_2 - x_1 \tag{5.5}$$

$$d_y = y_2 - y_1 \tag{5.6}$$

Ist $dx > dy$, so verläuft die Hauptrichtung der Linie entlang der X-Achse. Der Bresenhamalgorithmus geht somit immer einen Schritt Richtung X-Achse, und wenn ein bestimmter Fehler e_i nicht negativ geworden ist, zusätzlich einen Schritt entlang der Y-Achse. Dieser Fehlerwert wird initialisiert mit:

$$e_0 = 2\,d_y - d_x \tag{5.7}$$

Der Fehlerwert ändert sich mit jedem Schritt nach:

$$e_i = \begin{cases} e_{i-1} - 2\left(d_x - d_y\right), & \text{wenn } e_{i-1} \geq 0 \\ e_{i-1} + 2d_y, & \text{sonst} \end{cases} \tag{5.8}$$

Der Standardbresenhamalgorithmus hat als wesentlichen Eingabeparameter einen Start- und einen Endpunkt, aus denen er dann die Haupt- und Fehlerrichtung bestimmt. Da die Linie nicht auf einer unendlichen planaren Ebene, sondern in einem zweidimensionalen Ringpufferspeicher gezeichnet wird, müssen die Speichergrenzen beachtet werden.

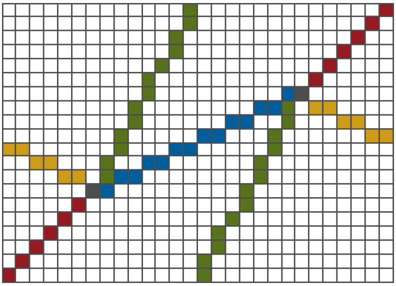


Abbildung 5.2.: Insgesamt ergeben sich vier verschiedene Möglichkeiten zwei Punkte (grau) in einem zweidimensionalen Ringpuffer zu verbinden

Dies erfolgt analog zum Punktalgorithmus mithilfe des Modulo-Operators, welcher einen Speicherüber- oder -unterlauf verhindert. Je nach aktueller Verschiebung ergeben sich im Ringpufferspeicher durch die Modulo-Operation vier Möglichkeiten eine Linie zwischen zwei Punkten zu zeichnen, welche in Abbildung 5.2 dargestellt sind.

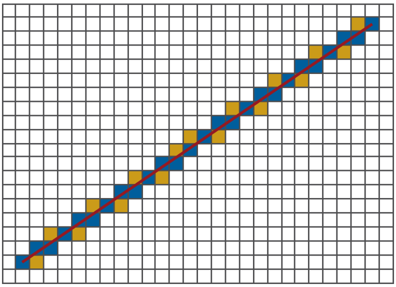


Abbildung 5.3.: Originale Variante des Bresenhamalgorithmus (blau) und dessen Erweiterung (gelb)

Der vorgestellte Bresenhamalgorithmus hat die Eigenschaft, dass nur

solche Zellen markiert werden, welche wesentlich auf der Linie liegen. Eine Modifikation wird in [Bouzouraa 2009] verwendet, in der alle Zellen markiert werden, die von der Linie auch nur marginal geschnitten werden. Damit wird insgesamt eine andere Diskretisierung erreicht, welche die Menge aller möglichen Zellen erhöht (vgl. Abbildung 5.3).

Je nach Verwendungszweck ist es sinnvoll, die eine oder andere Variante zu wählen: Falls mit der Linie ein Belegtbereich markiert soll oder die Befahrbarkeit eines Bereiches analysiert werden soll, so garantiert Bouzouraas Ansatz, dass durch Diskretisierungsungenauigkeiten keine Zellen unbeachtet gelassen werden, welche nur teilweise auf der Linie liegen. Falls jedoch ein Freibereich durch die Linie markiert werden soll, so ermöglicht der klassische Ansatz von Bresenham, dass nur solche Zellen als frei markiert werden, welche auch wesentlich von der Linie geschnitten werden.

5.3.4. Entwicklung eines Polygonalgorithmus

Viele Sensoren liefern nach Messvorverarbeitungsschritten neben der Position von Objekten auch deren Ausrichtung und Ausdehnung. Diese weisen eine Länge und Breite auf und können somit einen ganzen Bereich im Belegungsgitter abdecken. Auch Freibereiche können sich über größere Areale erstrecken und auch der prognostizierte Fahrschlauch des Ego-Fahrzeuges bildet eine größere Fläche. Im Folgenden wird deshalb ein Algorithmus benötigt, der flexibel alle Zellen zurückliefert, welche sich unter einer beliebig großen definierten Fläche befinden.

Anforderungen

Für den Zugriff auf Zellen solcher Flächen ist zunächst eine Beschreibung dieser Flächen notwendig. Das allgemeine Polygon eignet sich insbesondere, da jeder zusammenhängende Bereich durch ein solches beliebig genau approximiert werden kann. Eine Reduktion auf konvexe Polygone scheint ungeeignet, da unregelmäßige Freibereiche und gekrümmte Fahrschläuche naturgemäß nur durch konkave Polygone approximiert werden können. Letztere Annahme macht die Entwicklung eines entsprechenden Algorithmus deutlich anspruchsvoller.

Als weitere Anforderung soll es problemlos möglich sein, jede Art von Polygonen abzufragen. Dazu gehören zum einen solche, die über das Belegungsgitter hinausragen. Auch in diesem Fall soll der Algorithmus alle Zellen zurückliefern, die das Polygon im Belegungsgitter überdeckt. Zum

anderen sollen auch entartete Polygone möglich sein, da die auf Zellen diskretisierten Polygone unter Umständen sich selbst berühren oder schneiden könnten. Dies kann verschiedene Gruppen von inneren aber unverbundenen Zellen zur Folge haben. Auch ist der Fall denkbar, dass Zellen im Inneren der diskretisierten Kontur, aber trotzdem außerhalb des Polygons liegen.

Im Folgenden soll ein Algorithmus entwickelt werden, welcher all diese Anforderungen vereint. Zusätzlich kommen noch die allgemeinen Anforderungen für eine Sensordatenfusion zum Tragen. Hier ist insbesondere die Echtzeitfähigkeits- und Modularitätsanforderung zu nennen. Der Algorithmus soll somit möglichst effizient und nicht mit speziellen Fusions- oder Abfragealgorithmen fest gekoppelt sein.

Ablauf des Gesamtalgorithmus

Der entwickelte Algorithmus ist in verschiedene Schritte unterteilt. Zunächst wird geprüft, ob sich das Polygon vollständig im Belegungsgitter befindet. Falls nicht, werden die Ecken des Polygons soweit korrigiert, dass sie sich auf einem Rand des Belegungsgitters befinden. Dazu werden die Ränder mit den Kanten des Polygons geschnitten, die das Belegungsgitter verlassen. Als Ergebnis erhält man ein neues Polygon, welches sich vollständig im Belegungsgitter befindet.

Als Nächstes wird mittels Bresenham-Algorithmus die Polygonkontur gezeichnet. Dieses erfolgt in einem extra Speicher, da der spätere Füllalgorithmus für beliebige Nachbarzellen verschiedene Typen von Traversierungsmarkierungen unterscheiden muss. Diese Traversierungsmarkierungsinformation könnte in den Zellen des Belegungsgitters abgelegt werden. Analog zum vorgestellten Linienalgorithmus würde dies aber eine besondere Behandlung des Ringpuffers mit diversen Fallunterscheidungen nach sich ziehen und die Komplexität des Algorithmus unnötig vergrößern. Um dies zu vermeiden, wird ein neues zweidimensionales Feld genutzt, welches zwar genauso viele Einträge wie das Belegungsgitter besitzt, aber kein Ringpufferspeicher darstellt. Nach Abschluss der Fülloperation wird das Ergebnis mittels Modulo-Operator (vgl. Abschnitt 5.3.1) auf den Ringpufferspeicher übertragen.

Um die erstellte Polygonkontur wird zunächst ein virtuelles begrenzendes Rechteck erstellt, welches einen Pixel Abstand zur Polygonkontur hat. Die Begründung dazu liefert der folgende Füllalgorithmus: Anstatt nur das Innere des Polygons zu füllen, wird auch der Bereich zwischen Polygonkontur und begrenzendem Rechteck gefüllt. Hintergrund ist, dass der auf ganzzahlige Operationen Füllalgorithmus pro Pixel sehr schnell ist, dieser

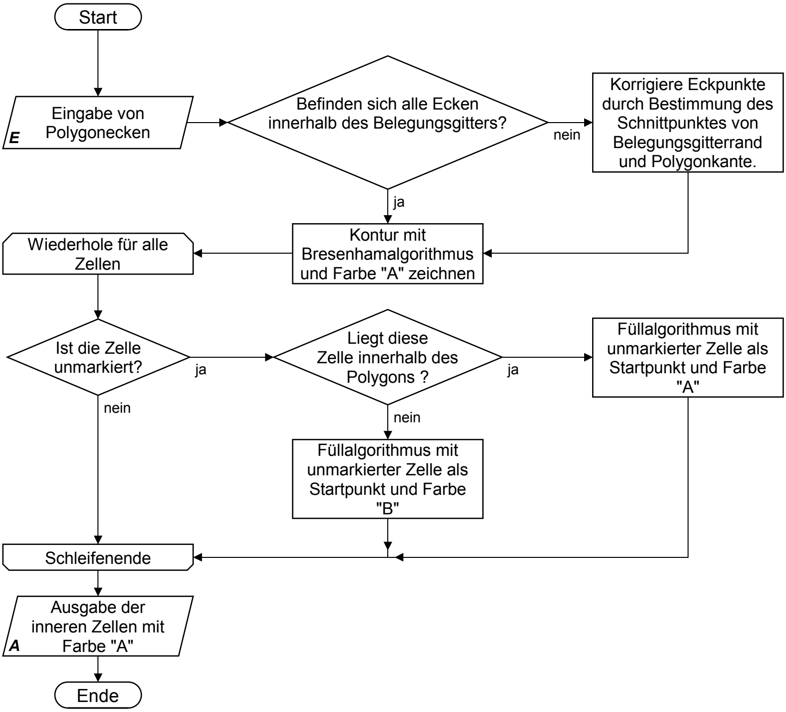


Abbildung 5.4.: Flussdiagramm des Polygonalgorithmus

jedoch einen Startpunkt (engl. „Seed Point“) benötigt. Das Bestimmen eines geeigneten Startpunktes innerhalb eines beliebigen konkaven und evtl. selbstschneidenen Polygons kann sehr rechenaufwendig sein: Der entsprechende Algorithmus, welcher im nächsten Abschnitt vorgestellt wird, basiert auf Gleitkommaoperationen und ist pro Pixel deutlich langsamer als der Füllalgorithmus. Um eine lange und eventuell sogar vergebliche Suche nach einem solchen Punkt zu vermeiden, wird der Füllalgorithmus verwendet, um zusätzlich den äußeren Bereich zu füllen und somit die Anzahl potenzieller Startpunkte für den inneren Bereich massiv zu reduzieren.

Insgesamt wird für jede Menge verbundener unmarkierter Zellen genau eine Zelle herausgegriffen und festgestellt, ob diese sich innerhalb oder außerhalb des Polygons befindet. Um die Anzahl solcher Mengen zu reduzieren, wurde ein Abstand für das begrenzende Rechteck zum Polygon

festgelegt. Somit müssen für die meisten Polygone nur zwei Startpunkte klassifiziert werden.

Sind sämtliche Zellen markiert, so werden die im Belegungsgitter korrespondierenden inneren Zellen des Polygons zurückgegeben. Wie beim Linienalgorithmus ist es auch möglich, optional die Zellen mit auszuwählen, die nur teilweise vom Polygon überdeckt werden.

In Abbildung 5.4 ist der Ablauf des Gesamtalgorithmus als Flussdiagramm dargestellt. Die Bestimmung, ob ein Punkt innerhalb oder außerhalb des Polygons liegt und der Füllalgorithmus selbst werden in den folgenden Abschnitten beschrieben.

Bestimmung eines Startpunktes für den Füllalgorithmus

Für den im folgenden Abschnitt beschriebenen Füllalgorithmus wird als Parameter ein Startpunkt benötigt. Der Füllmarkierungstyp ist abhängig davon, ob sich dieser innerhalb oder außerhalb des Polygons befindet.

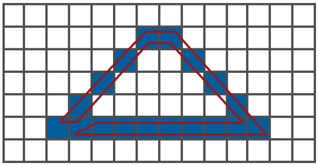


Abbildung 5.5.: Die dreieckige diskretisierte Polygonumrandung (blau) suggeriert ein einfaches Dreieck. Das zugrunde liegende Polygon weist die scheinbar inneren Zellen jedoch als äußere aus (roter Linienzug).

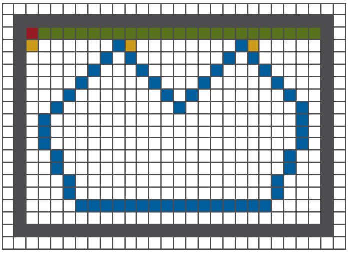
Zur Bestimmung, ob ein Punkt innerhalb oder außerhalb des Polygons liegt, ist es notwendig, das Polygon als Ganzes zu betrachten. Nur aufgrund der mittels Bresenhamalgorithmus erzeugten Umrandung ist eine solche Klassifizierung im Allgemeinen unentscheidbar, was Abbildung 5.5 verdeutlicht.

Um zu ermitteln, ob ein Punkt mit den Koordinaten x und y innerhalb eines n-seitigen Polygones mit den Eckpunkten $P_n = [(x_1, y_1), \ldots, (x_n, y_n)]$ liegt, wird eine Methode nach [Haines 1994] verwendet, welche auf dem sog. Jordan-Curve-Theorem basiert: Hierbei wird ein semi-unendlicher Strahl beginnend bei (x, y) horizontal durch das Polygon gesendet, sodass y konstant bleibt. Es wird gezählt, wie oft der Strahl eine Kante des Polygons trifft. Bei ungerader Anzahl liegt der Punkt innerhalb des Polygons,

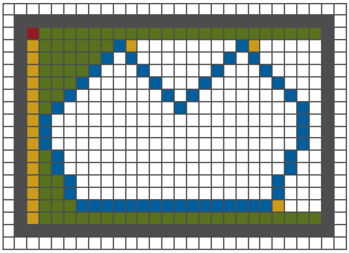
andernfalls außerhalb. Trifft der Strahl einen Eckpunkt, so ist dieser nur einfach zu zählen. Die Komplexität des Algorithmus ist linear zu der Anzahl der Ecken n des Polygons, sodass $O(n) = n$ gilt.

Füllalgorithmus

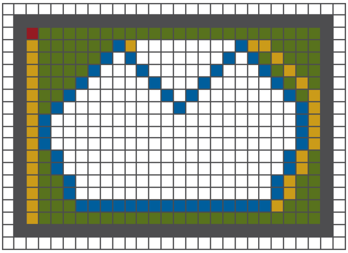
Der im vorangegangen Abschnitt vorgestellte Inklusionstest kann theoretisch direkt genutzt werden, um festzustellen, welche Zellen einer Zellmenge sich innerhalb eines Polygons befinden. Da dieser jedoch eine lineare Komplexität aufweist, wird in diesem Abschnitt eine effizientere Methode vorgestellt.



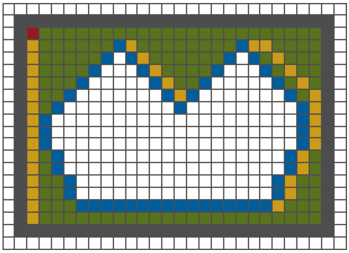
(a) Der Scanline-Algorithmus beginnt vom Startpunkt (rot) aus und setzt Füllpunkte (grün) und neue Startpunkte (gelb)



(b) Der Scanline-Algorithmus hat die ersten horizontalen Scans anhand der linken Startpunkte absolviert.



(c) Der Scanline-Algorithmus hat die rechten Startpunkte fertig bearbeitet. Die umgekehrte konvexe Hülle ist somit gefüllt.



(d) Der fehlende obere konkave Bereich ist jetzt fertig gefüllt.

Abbildung 5.6.: Eine Box (grau) umrandet das Polygon (blau) mit einer Zelle Abstand, bevor der Füllalgorithmus startet.

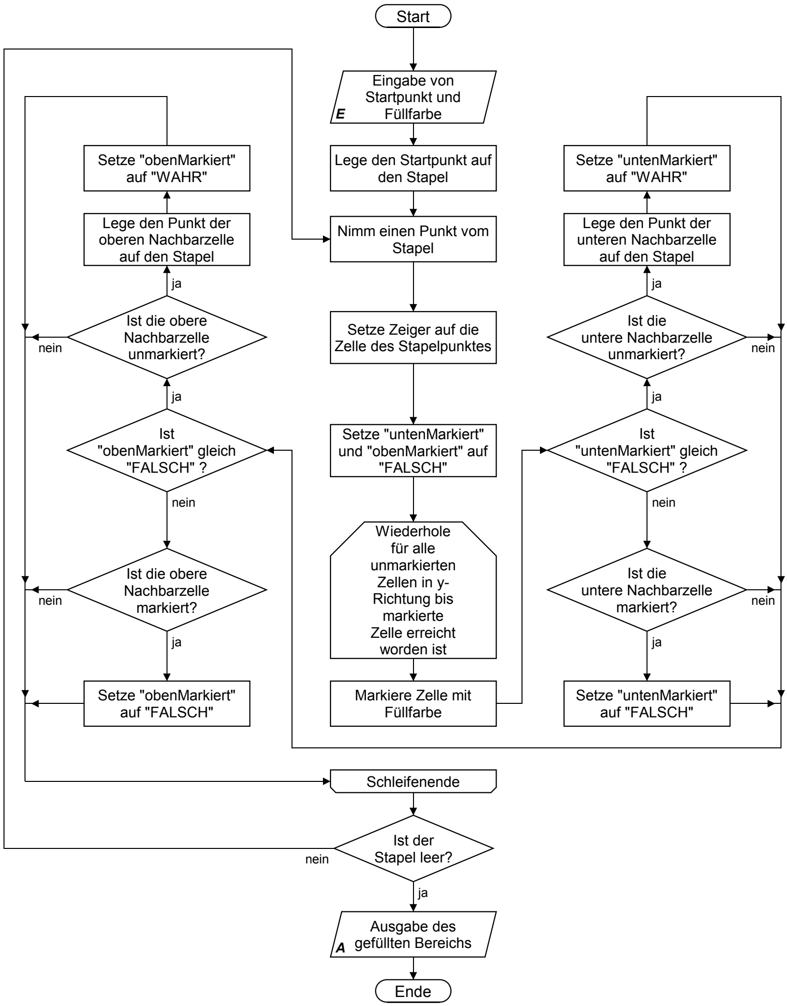


Abbildung 5.7.: Flussdiagramm des Füllalgorithmus.

Hierzu wird ein Füllalgorithmus verwendet, welcher die Aufgabe hat, ausgehend von einer Startzelle alle verbundenen Zellen zu markieren. Konkret wird im Folgenden eine Variation des in [Heckbert 1990] vorgestellten Algorithmus genutzt, welcher in Abbildung 5.6 beispielhaft visualisiert wird. Da der Füllalgorithmus möglichst effizient ablaufen soll, wurde er als zeilenbasierter iterativer Ansatz implementiert.

Der Algorithmus nutzt einen Stapel als Datenstruktur, auf dem zunächst der Startpunkt abgelegt wird. Insgesamt wird der Stapel solange mit einem Subalgorithmus abgearbeitet, bis dieser leer ist. Der Subalgorithmus nimmt als Parameter einen Punkt vom Stapel und sucht links und rechts nach Begrenzungen in Form von markierten Zellen. Zwischen diesen Begrenzungen werden die Zellen mit der gewünschten Füllfarbe markiert. Nebenbei werden die oberen und unteren Nachbarzellen analysiert und bei Erfüllung bestimmter Kriterien neue Startpunkte auf den Stapel geschrieben. Hier wird genau ein neuer Startpunkt für eine leere Nachbarzeile eingefügt. Ist diese Nachbarzeile jedoch durch Konturmarkierungen unterbrochen, was bei beispielsweise konkaven Polygonen möglich ist, so erkennt der Algorithmus dies und erzeugt entsprechend mehr Startpunkte für diese Zeile. Dieses ist im Beispiel von Abbildung 5.6a der Fall, wo insgesamt drei Startpunkte für eine Zeile neu hinzukommen. Für einen genauen Ablauf des Füllalgorithmus sei auf das Flussdiagramm in Abbildung 5.7 verwiesen.

Der Füllalgorithmus ist offensichtlich unabhängig von der Anzahl der verwendeten Polygoneckpunkte, sodass eine konstante Komplexität pro Pixel gegeben ist. Es gilt somit $O(n) = 1$, was den Algorithmus formal weniger komplex macht als den Startpunktalgorithmus. Des Weiteren benutzt er keine Fließkommaarithmetik, was den Algorithmus insgesamt sehr effizient macht.

5.4. Grundlagen: Logikalgorithmen für Belegungsgitter

Im vorherigen Abschnitt sind Algorithmen vorgestellt worden, welche geeignet sind, spezifische Zellen aus einem Belegungsgitter zu extrahieren. Mithilfe dieses Ansatzes können beispielsweise Zellen identifiziert werden, welche mit einer Messung korrespondieren.

Um solche Zellen sinnvoll zu aktualisieren, sind sog. probabilistische Logiken hilfreich. Drei solcher Logikalgorithmen werden in diesem Abschnitt dargestellt, welche unterschiedliche Zustandsrepräsentationen integrieren. Diese Logikalgorithmen bilden außerdem das Grundgerüst für verschiedene andere Fusionsalgorithmen, welche im Laufe des Kapitels vorgestellt werden.

5.4.1. Wahrnehmungsrahmen und Massefunktionen

Die hier vorgestellten Logiken erlauben es, mehrere Zustände entsprechend eines mathematischen Modells zu einem Zustand zusammenzufassen. Diese Algorithmen betrachten dabei einen sog. Wahrnehmungsrahmen (engl. „frame of discernment“) Θ, der verschiedene mögliche Hypothesen über einen Zellzustand beinhaltet (vgl. [Shafer 1976]). Dieser wird im Folgenden insoweit vereinfacht, als dass er auf nur zwei Basishypothesen reduziert wird. Diese unterscheiden, ob eine Zelle frei (H_F) oder belegt (H_B) ist:

$$\Theta = \{H_F, H_B\} \tag{5.9}$$

Aus dem Wahrnehmungsrahmen wird eine Ereignismenge Ω abgeleitet. Diese unterscheidet sich je nach Logik, bildet aber jeweils einen vollständigen Ereignishorizont. Auf Basis von Ω wird im Folgenden eine sog. „Massefunktion“ (engl. „mass function“) P definiert (vgl. [Shafer 1976]), für die gilt:

$$P(H_\varnothing) = 0 \tag{5.10}$$

$$P(H_X) \in [0,1], \forall X \in \Omega \tag{5.11}$$

$$\sum_{\forall X \in \Omega} P(H_X) = 1 \tag{5.12}$$

Die Ergebnismenge der Funktion $P(X)$ heißt dann Massenverteilung[2]. Bei vielen Ansätzen sind solche Massenverteilungen sowohl das Ergebnis inverser Sensormodelle für einzelne Messungen m_i bzgl. bestimmter Zellen als auch für die fusionierten Zellzustände c_k.

Im Allgemeinen wird ein solcher Zellzustand c_k, der auf dem Initialzustand z_0 und einer Menge von Messwerten $M^k = \{m_0, \ldots, m_k\}$ basiert, aufgrund einer Messung m_{k+1} zu einem Zustand c_{k+1} aktualisiert. Viele Algorithmen basieren auf einem Markov-Prozess erster Ordnung:

$$P(c_{k+1}|M^{k+1}) = P(c_{k+1}|c_k, m_{k+1}) \tag{5.13}$$

Der Folgezustand c_{k+1} hängt somit nur vom vorherigen Zustand c_k und der Messung m_{k+1} ab. Alle vorangegangenen Messungen M_k sind bereits im Zustand c_k berücksichtigt.

[2] Zur Unterscheidung von anderen Tupeln werden Massenverteilungen im Folgenden durch Tupel der Form $P(X) = \langle \ldots \rangle$ dargestellt

Welche Attribute der Zustand c_k aufweist, hängt von der verwendeten Logik und der daraus resultierenden Massenverteilung ab. Drei verschiedene Logiken werden in diesem Abschnitt vorgestellt, welche beispielhaft versuchen, freie von belegten Zellen möglichst genau zu unterscheiden.

5.4.2. Bayes-Logik

Beim Bayesansatz entspricht die Ereignismenge Ω_{Bayes} dem Wahrnehmungsrahmen Θ. Die Massenverteilung einer Zelle lässt sich somit mit dem Tupel $\langle P(H_F|M^k), P(H_B|M^k)\rangle$ beschreiben[3]. Die Elemente des Tupels enthalten die Frei- und Belegtwahrscheinlichkeiten dieser Zelle, welche aufgrund der Messungen M^k gebildet worden sind. Diese beiden Wahrscheinlichkeiten sind gemäß der Massefunktionsdefinition komplementär zu betrachten.

Initial wird meist davon ausgegangen, dass die Zelle mit gleicher Wahrscheinlichkeit belegt bzw. frei ist und somit $\langle P(H_F|M^0), P_0(H_B|M^0)\rangle = \langle 1/2, 1/2\rangle$ ist. Dieser Zustand repräsentiert die maximale Entropie, die die Zelle erreichen kann, welches gleichzeitig der maximalen Unsicherheit entspricht. Vorteilhaft an dieser Annahme ist außerdem, dass für die Berechnung des sog. „Likelihood-Quotient“ bzw. „ Bayes-Faktor“ $LR(M^k)$ bereits zwei Parameter feststehen. Dieser ergibt sich nach dem Bayes-Theorem zu:

$$LR(M^k) = \frac{P(H_F|M^k)}{P(H_B|M^k)} = \frac{\frac{P(M^k|H_F)\cdot P(H_F)}{P(M^k)}}{\frac{P(M^k|H_B)\cdot P(H_B)}{P(M^k)}} \tag{5.14}$$

$$= \frac{P(M^k|H_F)}{P(M^k|H_B)} \cdot \frac{P(H_F)}{P(H_B)} \tag{5.15}$$

Für die Berechnung des „Likelihood-Quotienten“ benötigt man somit die Detektionswahrscheinlichkeiten der Frei- und Belegtbereiche und deren Initialwahrscheinlichkeiten. Diese Parameter lassen sich aus Ground Truth-Daten bestimmen. Die Auftretenswahrscheinlichkeit der Messung $P(M^k)$ muss nicht bestimmt werden, da sich diese herauskürzt.

Der Zellzustand wird rekursiv aktualisiert: Sei $\langle P(m_{k+1}|H_F), P(m_{k+1}|H_B)\rangle$ die Massenverteilung einer neuen Messung m_{k+1} gemäß inversen Sensormodells für eine Zelle. Deren Zellzustand $\langle P(H_F|M^k), P(H_B|M^k)\rangle$ aktualisiert sich dann mit:

[3] Hinsichtlich einer einheitlichen Beschreibung aller Logikalgorithmen wird hier eine alternative Darstellung des Bayes-Algorithmus gewählt.

$$P(H_F|M^{k+1}) = \left(LR(M^k)^{-1} \cdot \frac{P(m_{k+1}|H_B)}{P(m_{k+1}|H_F)} + 1\right)^{-1} \tag{5.16}$$

$$P(H_B|M^{k+1}) = 1 - P(H_F|M^{k+1}) \tag{5.17}$$

Beträgt die Massenverteilung einer Messung $\langle 1/2, 1/2 \rangle$, so ändern die Gleichungen 5.16 und 5.17 den Zellzustand nicht, da diese sowohl Hypothese (Zelle ist frei) als auch Gegenhypothese (Zelle ist belegt) gleichermaßen stützt. Diese Massenverteilung bildet somit das neutrale Element.

Der Bayes-Algorithmus ist sowohl assoziativ als auch kommutativ. Somit spielt es für das Ergebnis keine Rolle, in welcher Reihenfolge die Messungen in den Zellzustand eingebracht werden.

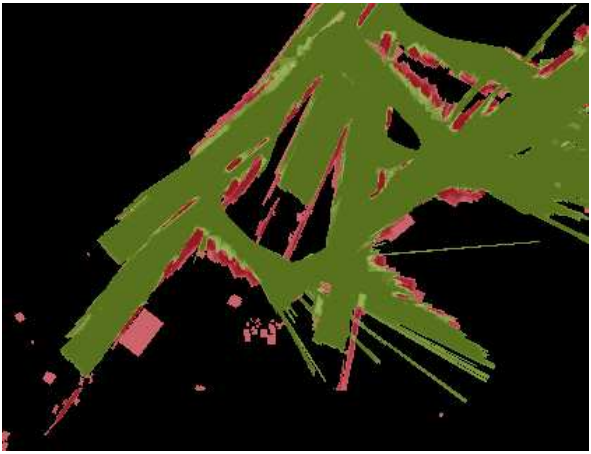


Abbildung 5.8.: Belegungsgitter als Ergebnis einer Parkplatzrundfahrt. Bayes-Algorithmus zeigt belegte und freie Bereiche in rot bzw. grün. Schwarzen Bereichen konnte nie eine Messung zugeordnet werden.

Ein Anwendungsbeispiel des Bayes-Algorithmus ist in Abbildung 5.8 dargestellt. Es zeigt ein Belegungsgitter, welches auf einem Parkplatz entstanden ist. Die Messdaten sind im Winter aufgenommen worden und sind durch Autos, Büsche, Bäumen, aufgeschüttete Schneeberge und vereiste Flächen beeinflusst. Dieses heterogene Umfeld enthält somit viele Bereiche, in der unklare Objektkonturen, Fehldetektionen und fehlerhafte Nichtdetektionen möglich sind.

Gemessene Bereiche, deren Belegtwahrscheinlichkeit größer als 1/2 ist, sind rot gefärbt, andernfalls sind sie frei und grün eingefärbt. Nie gemessene Bereiche sind schwarz dargestellt.

Bei der Messung wurde eine Eigenbewegungsschätzung aufgrund der Raddrehzahl- und Gierratensensoren erzeugt (vgl. Abschnitt 3.5.5), welche einen sich integrierenden Fehler aufweist. Je länger die zurückgelegte Strecke auf einer Rundfahrt ist, desto ungenauer werden die neuen Messdaten auf das Belegungsgitter abgebildet. Diese Schätzung bildet somit eine weitere potenzielle Fehlerquelle für die ungenaue örtliche Zuordnung einer Messung.

5.4.3. Dempster-Shafer Theorie

Die vorgestellte Bayes-Logik hat den Nachteil, dass selten gemessene Zellen nicht von solchen Zellen unterschieden werden können, die im Laufe des Fusionsprozesses widersprüchliche Daten bekommen haben. Es fehlt eine Angabe, wie vage die Schätzung ist.

Diesen Nachteil kann man durch die Erweiterung der sog. „Evidenztheorie“ von Dempster und Shafer (DST) begegnen (vgl. [Shafer 1976]). Diese erweitert die Ereignismenge Ω um zusätzliche Elemente, sodass man explizit Unwissen in einer Massenverteilung angeben kann. Diese Elemente werden aus allen möglichen Vereinigungsmengen des Wahrnehmungsrahmens gebildet:

$$\Omega_{DST} = \{\forall X \subseteq \Theta : P(H_X)\} \tag{5.18}$$

Elemente aus Ω_{DST}, die aus der Vereinigung von mehreren nichtleeren Elementen des Wahrnehmungsrahmens entstanden sind, repräsentieren eingetretene unsichere Ereignisse. Ein solches unsicheres Ereignis kann auf mehrere Hypothesen aus dem Wahrnehmungsrahmen basieren.

Für den vereinfachten zweielementigen Wahrnehmungsrahmen ergibt sich nur ein einziges weiteres Element für die Ereignismenge Ω_{DST}: Neben H_F und H_B, die den Frei- bzw. Belegtzustand darstellen, beinhaltet es nun noch das Ereignis $H_{F\cup B}$. Dieses repräsentiert die vollständige Zustandsunsicherheit, indem eine Zelle gleichzeitig als „frei“ oder „belegt“ deklariert wird.

Liegt eine Messung vor, die eine Belegt- oder Freihypothese für die Zelle liefert, so wird der Zellzustand wie folgt aktualisiert: Sei $\left\langle P(H_F|M^k), P(H_B|M^k), P(H_{F\cup B}|M^k)\right\rangle$ eine Massenverteilung, die den alten Zellzustand repräsentiert. Dieser beinhaltet eine Freievidenz, eine Belegtevidenz und eine Evidenz für den unsicheren Zustand. Diese wird durch

eine Massenverteilung $\langle P(m_{k+1}|H_F), P(m_{k+1}|H_B), P(m_{k+1}|H_{F\cup B})\rangle$ basierend auf einer Messung m_{k+1} aktualisiert. Zunächst wird aus diesen Variablen der Konfliktwert K ermittelt:

$$K = P(H_F|M^k) \cdot P(H_B|m_{k+1}) + P(H_B|M^k) \cdot P(H_F|m_{k+1}) \tag{5.19}$$

Der Konfliktwert gibt an, wie stark sich die beiden Informationsquellen widersprechen. Da er auf kein Element der Ereignismenge Ω_{DST} komplett abgebildet werden kann, wird er herausnormalisiert und somit proportional auf die Elemente der Ereignismenge verteilt. Der neue Zellzustand $\left\langle P(H_F|M^{k+1}), P(H_B|M^{k+1}), P(H_{F\cup B}|M^{k+1})\right\rangle$ ergibt sich dann zu:

$$\begin{aligned}
P(H_F|M^{k+1}) &= \frac{1}{1-K} \cdot \Big[P(H_F|M^k) \cdot P(H_F|m_{k+1}) \\
&\quad + P(H_{F\cup B}|M^k) \cdot P(H_F|m_{k+1}) \\
&\quad + P(H_F|M^k) \cdot P(H_{F\cup B}|m_{k+1})\Big] && (5.20) \\
P(H_B|M^{k+1}) &= \frac{1}{1-K} \cdot \Big[P(H_B|M^k) \cdot P(H_B|m_{k+1}) \\
&\quad + P(H_{F\cup B}|M^k) \cdot P(H_B|m_{k+1}) \\
&\quad + P(H_B|M^k) \cdot P(H_{F\cup B}|m_{k+1})\Big] && (5.21) \\
P(H_{F\cup B}|M^{k+1}) &= \frac{1}{1-K} \cdot P(H_{F\cup B}|M^k) \cdot P(H_{F\cup B}|m_{k+1}) && (5.22)
\end{aligned}$$

Wie beim Bayes-Algorithmus ist das neutrale Element das Element, welches die höchste Entropie aufweist und dem Initialzustand gleicht. Dieses hat keine Information über Belegt- und Freiwahrscheinlichkeiten und weist eine maximale Unwissenheit mit $\langle P(H_F), P(H_B), P(H_{F\cup B})\rangle = \langle 0, 0, 1\rangle$ auf.

Auch der Dempster-Shafer-Algorithmus ist assoziativ und kommutativ, sodass der Zellzustand von der Reihenfolge der eingebrachten Messungen unabhängig ist.

Das Parkplatzbeispiel aus dem vorherigen Abschnitt ist auch mit der Dempster-Shafer Theorie getestet worden. Abbildung 5.9 zeigt das Ergebnis. Zusätzlich zu den freien und belegten Bereichen gibt es nun noch graue Bereiche. Diese wurden zwar durch Messungen aktualisiert, jedoch dominiert hier noch die Unbekannt-Masse die Zellschätzung.

Der Vergleich der beiden Abbildungen zeigt, dass beim Bayesansatz mehr Bereiche existieren, die als frei oder belegt deklariert werden, da dieser keine Mindestevidenzmasse sammeln muss.

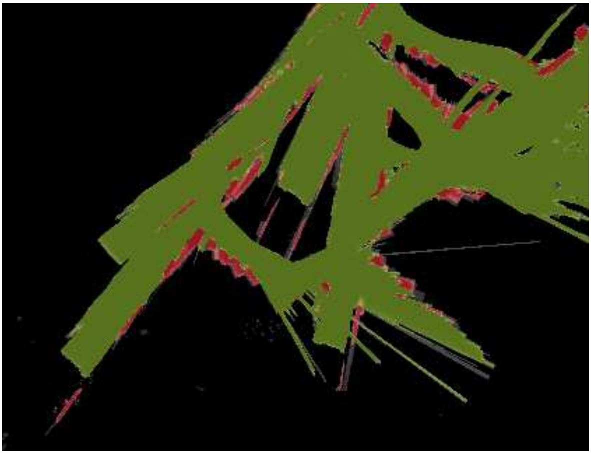


Abbildung 5.9.: Erzeugtes Belegungsgitter nach einer Parkplatzrundfahrt. Der Dempster-Shafer-Algorithmus zeigt belegte, freie und unbekannte Bereiche in rot, grün bzw. grau. Schwarze Bereiche sind Bereiche mit maximalen Unwissen.

5.4.4. Dezert-Smarandache Theorie

Die vorgestellte Evidenztheorie von Dempster-Shafer berücksichtigt Unwissen bereits in einem eigenen Zustand. Eine explizite Modellierung von widersprüchlichen Informationen fehlt jedoch. Diese werden zwar im vorgestellten Aktualisierungsalgorithmus in Gleichung 5.19 mitberechnet, aber letztendlich herausnormalisiert.

Dieser fehlende Aspekt wurde in der Evidenztheorie von Dezert-Smarandache (DSmT) erweitert (vgl. [Smarandache 2006]). Diese berücksichtigt explizit in zusätzlichen Attributen den Anteil an widersprüchlichen Informationen.

Während bei der Dempster-Shafer Theorie die Ereignismenge Ω_{DST} aus allen möglichen Vereinigungsmengen des Wahrnehmungsrahmens besteht, erweitert die Dezert-Smarandache Theorie diese um alle möglichen Schnittmengen. Die Ereignismenge Ω_{DSmT} entsteht somit mittels:

$$\Omega_{DSmT} = \{\forall X, Y \subseteq \Theta : H_{X \cap Y}\} \tag{5.23}$$

Das Tupel zur Modellierung von Zellzuständen und Messungen wird bei dem vereinfachten Wahrnehmungsrahmen um ein einzelnes Konflikt-

wertattribut erweitert. Sowohl im Initialzustand als auch in den einzelnen Messungen sollte dieses null betragen, da beides atomare und sich nicht selbst widersprechende Informationen repräsentieren. Die Massenverteilung $\langle 0,0,1,0 \rangle$ bildet somit das neutrale und initiale Element für die Dezert-Smarandache Theorie.

Der Zellzustand $\left\langle P(H_F|M^k), P(H_B|M^k), P(H_{F\cup B}|M^k), P(H_{F\cap B}|M^k) \right\rangle$ wird nach der Dezert-Smarandache Theorie wie folgt aktualisiert:

$$\begin{aligned} P(H_F|M^{k+1}) &= P(H_F|M^k)P(H_F|m_{k+1}) + P(H_F|M^k)P(H_{F\cup B}|m_{k+1}) \\ &\quad + P(H_{F\cup B}|M^k)P(H_F|m_{k+1}) \qquad (5.24) \\ P(H_B|M^{k+1}) &= P(H_B|M^k)P(H_B|m_{k+1}) + P(H_B|M^k)P(H_{F\cup B}|m_{k+1}) \\ &\quad + P(H_{F\cup B}|M^k)P(H_B|m_{k+1}) \qquad (5.25) \\ P(H_{F\cup B}|M^{k+1}) &= P(H_{F\cup B}|M^k)P(H_{F\cup B}|m_{k+1}) \qquad (5.26) \\ P(H_{F\cap B}|M^{k+1}) &= 1 - P(H_F|M^{k+1}) - P(H_B|M^{k+1}) - P(H_{F\cup B}|M^{k+1}) \qquad (5.27) \end{aligned}$$

Der Algorithmus liefert eine sehr genaue Repräsentation eines Zellzustandes mit eventuell fehlenden bzw. widersprüchlichen Informationen. Für die Praxis hat er jedoch zwei Nachteile.

Zum einen ist er zwar assoziativ, aber nicht kommutativ, sodass die Messreihenfolge einen Einfluss auf das Ergebnis hat. Je nach Latenz der Sensoren können somit unterschiedliche Ergebnisse errechnet werden.

Als zweiter Nachteil ist zu nennen, dass der Konfliktwert monoton steigt. Hat man beispielsweise in großer Entfernung aufgrund von Messungenauigkeiten mehrere widersprüchliche Informationen, so kann eine Zelle schnell einen großen Konfliktwert nahe eins erreichen. Wenn sich das Fahrzeug der Zelle nähert, verringern sich die Messungenauigkeiten zu dieser. Somit steigt die Wahrscheinlichkeit, dass keine widersprüchliche Messungen mehr vorliegen. Dennoch kann der Konfliktwert nicht verringern werden.

In der Praxis müssen deshalb Strategien entwickelt werden, mit denen der Konfliktwert über die Zeit verringert werden kann. In zeitlich regelmäßigen Abständen kann man beispielsweise einen Teil der Konfliktmasse dem unbekannten Zustand zuschlagen und somit neue Messungen höher gewichten. Alternativ kann man wie bei der Dempster-Shafer Theorie diesen Konfliktmassenanteil herausnormalisieren und somit über die restlichen Attribute verteilen. Dieser Ansatz stärkt somit den alten Zellzustand und erhält die Proportionen der übrigen Attribute.

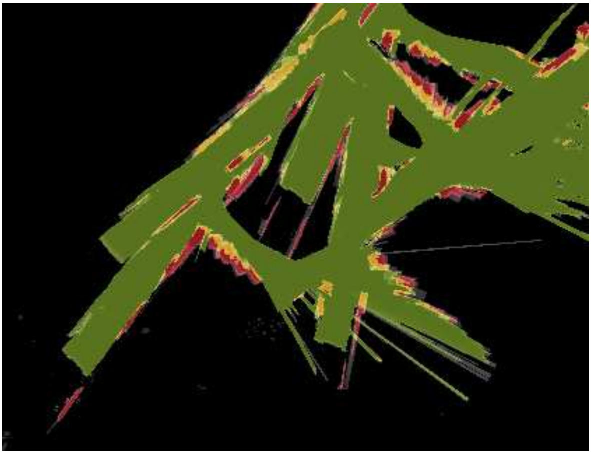


Abbildung 5.10.: Berechnetes Belegungsgitter nach Rundfahrt durch einen Parkplatz. Der Dezert-Smarandache-Algorithmus zeigt neben belegten, freien und unbekannten Bereichen (rot, grün bzw. schwarz/grau) auch Konfliktbereiche (gelb).

Ein weiterer Ansatz wird später im Abschnitt 5.7.2 vorgestellt. Dieser entfernt alte Messungen aus einer Zelle, sodass Konfliktwerte bei konsistenten neuen Daten verschwinden.

In Abbildung 5.10 ist die Dezert-Smarandache Theorie mit diesem Ansatz auf das Parkplatzbeispiel aus den vorherigen Abschnitten angewendet worden. Es zeigt neben den freien, belegten und unbekannten Bereichen auch eine Vielzahl von Konfliktbereichen (gelb). Diese sind mit den bereits erwähnten Odometriefehlern der Eigenbewegungsschätzung und der komplex strukturierten Domäne begründet. So sind auf dem Parkplatz viele Objekte zu finden, die keine klar abgegrenzte Kontur in Rechteckform aufweisen, wie beispielsweise Büsche und aufgeschüttete Schneeberge.

5.4.5. Zusammenfassung

In diesem Abschnitt sind aus der Literatur bekannte Logikalgorithmen vorgestellt worden, die freie von belegten Bereichen unterscheiden können. Im Sinne der Modularitätsanforderung sind diese einheitlich als Massefunktionen dargestellt worden, so dass diese leicht austauschbar sind. Diese bilden zusammen mit den Algorithmen und Datenstrukturen für Bele-

gungsgitter die Grundlage für die kommenden Abschnitte.

Eine Bewertung der vorgestellten Logiken ist im Speziellen nur zusammen mit einer spezifischen Anwendung möglich. So könnte man zum Beispiel die Dempster-Shafer Theorie einsetzen, wenn man selten oder ungemessene Bereiche identifizieren möchte, um diese auszuschließen. Eine Anwendung für die Dezert-Smarandache Theorie hingegen könnte die Eigendiagnose der Umfeldwahrnehmung sein: So könnten häufige und hohe Konfliktwerte beispielsweise auf eine defekte oder dekalibrierte Sensorik hindeuten. Für eine reine Darstellung der Befahrbarkeitswahrscheinlichkeit reicht jedoch die Bayes-Logik aus.

Die Bayes-Logik hat die Eigenschaft, dass diese A-priori-Wahrscheinlichkeiten als Parameter benötigt, welche jedoch unbekannt bzw. stark von variablen Faktoren abhängig sein können. Die Anwendung dieser Logik erfordert somit eine genauere Untersuchung des Zustandsraumes als bei der Dempster-Shafer Theorie oder der Dezert-Smarandache Theorie. Dafür liefert die Bayes-Logik bei ausbleibenden Messungen auch direkt A-priori-Wahrscheinlichkeiten und keine Unbekannt-Massen, was für Anwendungen eventuell sinnvoller ist.

Zur Darstellung der Massenverteilungen steigt mit zunehmender Elementanzahl der Ergebnismenge der Speicherbedarf, was besonders bei Belegungsgittern mit vielen Zellen ins Gewicht fallen könnte. So benötigt die komplementäre Darstellung von Bayes ein Attribut, von der Dempster-Shafer Theorie zwei Attribute und von der Dezert-Smarandache Theorie drei Attribute.

5.5. Stand der Technik: Fusionsalgorithmen für Belegungsgitter

Die im vorherigen Abschnitt vorgestellten Logiken sind in der Lage, Informationsquellen in Form von Massenverteilungen zu fusionieren. Hier fehlt jedoch noch ein Modell, wie man eine solche Massenverteilung für einzelne Messungen generiert. Dazu sind unter anderem inverse probabilistische Sensormodelle notwendig, die solche Massenverteilungen erzeugen und dabei beispielsweise Messungenauigkeiten und die Zuverlässigkeit des Sensors berücksichtigen.

Ein zweiter Aspekt beschäftigt sich mit der Form, in der das Belegungsgitter später genutzt wird. Je nach Anforderung der Anwendung könnte beispielsweise die probabilistische Befahrbarkeit eines gewählten Pfades bestimmt werden. Als Erweiterung ist es denkbar, dass ein Pfad automa-

tisch bestimmt werden soll, der möglichst viel Abstand zu belegten Zellen hält und somit die Befahrbarkeitswahrscheinlichkeit maximiert.

In diesem Abschnitt werden dazu verschiedene bekannte Ansätze vorgestellt und bewertet, wobei solche mit einem hohen Rechenaufwand hinsichtlich Echtzeitanforderungen ausgeschlossen wurden. Hierzu gehört beispielsweise das Erlernen von Belegungsgitterkarten [Thrun 2003] mittels nicht-inversen Sensormodellen und Expectation-Maximization-Algorithmus.

5.5.1. Belegungsgitterfusion nach Elfes

Die ersten Arbeiten zur Sensordatenfusion mittels Belegungsgittern sind die Arbeiten [Moravec 1985], [Elfes 1989a], [Elfes 1989b], [Elfes 1991] und [Elfes 1992] von Elfes und Moravec. Die verwendeten Sensorträger bilden Roboter, die sich im Gebäude als auch im freien Gelände zurechtfinden sollen. Diese sind mit Ultraschallsensoren ausgestattet, womit sie die nahe Umgebung erfassen können. Odometriesensoren ermöglichen eine Eigenbewegungsschätzung. Einer der genutzten Versuchsträger ist der Roboter „Neptune“, der in Abbildung 5.11 dargestellt ist.



Abbildung 5.11.: Verwendeter Versuchsträger „Neptune“ mit einem Array aus 24 Ultraschallsensoren und 2 Kameras (Bild aus [Moravec 1985])

Elfes nimmt an, dass die Messfehler der Ultraschallsensoren normalverteilt sind (vgl. [Elfes 1989b]). Dies gilt sowohl für die gemessene Entfernung

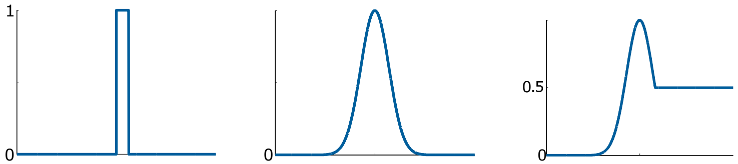
z als auch für den Winkel θ. Die WDF $p(H_B|r,z,\theta)$ modelliert somit für Zellen mit der Entfernung r eine zweidimensionale Normalverteilung hinsichtlich Polarkoordinaten mittels:

$$p(H_B|r,z,\theta) = \frac{1}{2\pi\sigma_r\sigma_\theta} e^{-\frac{1}{2}\cdot\left[\frac{(r-z)^2}{\sigma_r^2}+\frac{\theta^2}{\sigma_\theta^2}\right]} \tag{5.28}$$

Elfes nutzt als Aktualisierungsalgorithmus für die Zellzustände den im Abschnitt 5.4.2 vorgestellten Bayes-Algorithmus. Problematisch an der WDF ist, dass Zellen die weit hinter einem gemessenen Hindernis liegen als „frei“ deklariert werden, obwohl eigentlich keine Informationen von verdeckten Zellen vorliegen. Deshalb modifiziert Elfes die WDF in der Form, dass das Funktionsergebnis der WDF für Zellen hinter Hindernissen nie kleiner als $1/2$ werden kann.

$$P(H_B|r,z,\theta) = \begin{cases} p(H_B|r,z,\theta) & , \text{ z < r} \\ \max(1/2, p(r|z,\theta)) & , \text{ sonst} \end{cases} \tag{5.29}$$

Die mehrschrittige Aufbereitung einer Messung ist in Abbildung 5.12 dargestellt.



(a) Sensormessung

(b) Überführung der Messung mittels inversen Sensormodells

(c) Konvertierung der WDF für die Bayeslogik

Abbildung 5.12.: Aufbereitung einer Messung für den bayesschen Aktualisierungsalgorithmus nach [Elfes 1989b]

Elfes nutzt die Messungen nicht nur für die Abbildung der lokalen Umgebung, sondern erstellt gleichzeitig eine globale Belegungskarte. Hier spielen Odometriefehler eine sehr viel größere Rolle, da diese sich integrieren je stärker sich der Roboter bewegt. Aus diesem Grund verwendet Elfes ein mehrstufiges Verfahren.

Die allgemeine Strategie besteht darin, die Stillstandsphase des Roboters für die Erzeugung eines möglichst genauen lokalen Belegungsgitter

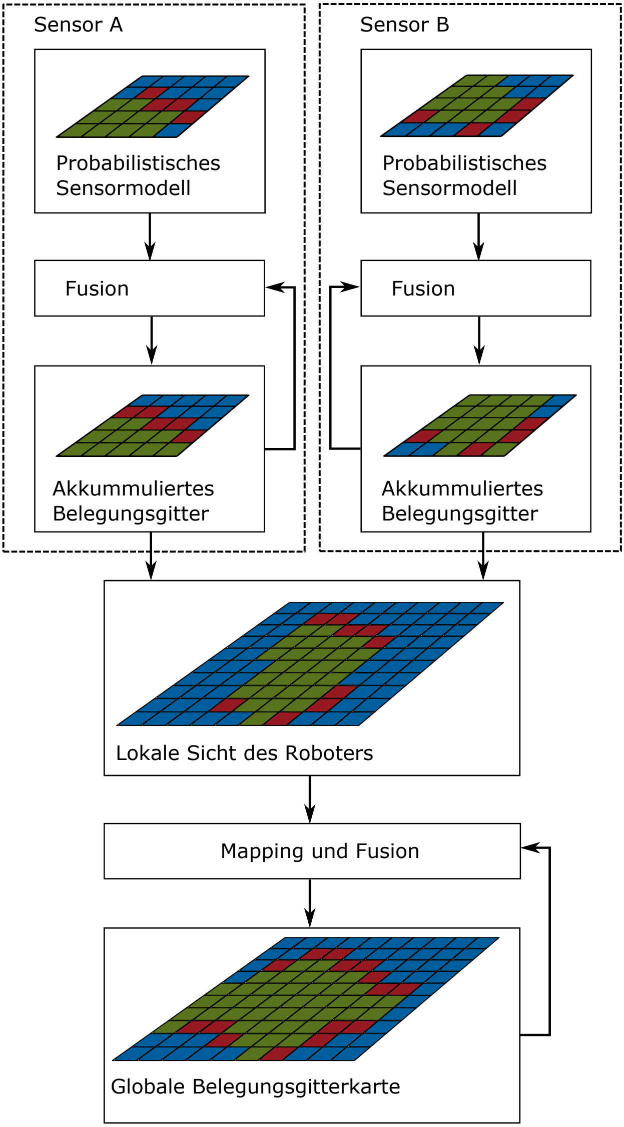


Abbildung 5.13.: Mehrschichtige Fusionsarchitektur für Belegungsgitter nach [Elfes 1989b]

zu nutzen, sodass Odometriefehler keinen Einfluss ausüben können. Kurz bevor der Roboter sich weiterbewegt, wird das lokale Belegungsgitter in ein Globales überführt.

Die genaue Vorgehensweise ist in Abbildung 5.13 dargestellt: Zunächst wird für eine einzelne Messung mittels des inversen probabilistischen Sensormodells ein lokales temporäres Belegungsgitter erzeugt. Dieses wird in ein zweites Gitter fusioniert, welches die akkumulierte Sicht eines einzelnen Sensors auf die lokale Umgebung repräsentiert.

Bevor der Roboter sich weiterbewegt, werden in einem dritten Schritt die unterschiedlichen Belegungsgitter der einzelnen Sensoren zu einem weiteren fusioniert. Bei dieser Transformation werden bekannte Einbaupositionen und Ausrichtungen der Sensoren berücksichtigt.

Das so gewonnene Belegungsgitter bildet die aggregierte Sicht des Roboters über die lokale Umgebung. Dieses wird in ein globales Belegungsgitter überführt. Elfes hat einen Alterungsmechanismus in das globale Belegungsgitter eingeführt, um länger unbeobachtete Bereiche schneller wieder auf die neue lokale Umgebung aktualisieren zu können.

Besonders schwierig hierbei ist die richtige Transformation zu finden, da sich die Roboterposition nur ungenau bestimmen lässt. Hierzu wird sowohl das Belegungsgitter der globalen Karte im vermuteten Aufenthaltsgebiet des Roboters als auch die lokale Robotersicht in eine geringe Auflösung transformiert. Als Nächstes werden geeignete Kandidaten für eine Transformation gesucht. Hierbei sollte die Differenz der Belegungsgitter möglichst gering sein. Eine mögliche Differenzberechnung ist in [Martin 1996] vorgeschlagen worden:

Sei C und D jeweils eine geordnete Menge von $n \times m$ Zellen, die verglichen werden sollen. Jede Zelle $c_{i,j} \in C$ bzw. $d_{i,j} \in D$ enthält die komplementären Wahrscheinlichkeiten für die Frei- und Belegthypothese H_F bzw. H_B. Das Ähnlichkeitsmaß $\gamma(C,D)$ liefert dann einen Gütewert im Intervall $[0,1]$ mit:

$$\gamma(C,D) = \prod_{i=1}^{n} \prod_{j=1}^{m} \left[P(H_F|c_{i,j}) \cdot P(H_F|d_{i,j}) + P(H_B|c_{i,j}) \cdot P(H_B|d_{i,j}) \right] \tag{5.30}$$

Volle Ähnlichkeit wird nur zurückgegeben, wenn $C = D$ und alle Zellen eine maximale Konfidenz aufweisen. Dies ist der Fall, wenn $P(H_B|c_{i,j}) \in \{0,1\}$ für alle Zellen $c_{i,j} \in C$ gilt. Um eine volle Unähnlichkeit zu erreichen, reicht bereits ein einziges ungleiches Paar von Zellen mit maximaler Konfidenz aus.

Diese extremen Konfidenzwerte werden bei der Verwendung von nichtidealen inversen Sensormodellen jedoch nie erreicht. Mit einer endlichen

Anzahl von Zellaktualisierungen kann die Bayes-Aktualisierungslogik ihre Grenzwerte null und eins zwar beliebig genau annähern, jedoch nie erreichen.

Bei der Produktbildung sehr vieler Faktoren aus dem Intervall [0, 1] können leicht numerische Ungenauigkeiten auftreten. Aus diesem Grund schlägt Elfes eine modifizierte logarithmische Darstellung vor. Dieser erlaubt einen summarischen Aktualisierungsalgorithmus, welcher eine höhere numerische Stabilität aufweist:

$$\begin{aligned}\Gamma(C,D) = &\quad m \cdot n + \log_2 \gamma(C,D) \\ = &\quad \sum_{i=1}^{n} \sum_{j=1}^{m} \left[1 + \log_2 \left(P(H_F|c_{i,j}) \cdot P(H_F|d_{i,j}) + P(H_B|c_{i,j}) \cdot P(H_B|d_{i,j})\right)\right]\end{aligned} \tag{5.31}$$

Durch das Gütemaß $\Gamma(C,D)$ wird eine Auswahl von geeigneten Transformationskandidaten getroffen. Diese werden weiter untersucht, indem die Auflösung der Belegungsgitter sukzessiv erhöht wird, bis die originale Auflösung erreicht worden ist. Der beste Kandidat liefert die zu verwendende Transformationsmatrix.

Nach der Erzeugung des globalen Belegungsgitters kann dieses als Entscheidungsgrundlage für die Pfadplanung dienen. Diese basiert bei Elfes im Wesentlichen auf dem A*-Algorithmus (vgl. [Hart 1968]), welcher in der Lage ist, den Pfad zwischen zwei Knoten in einem Graphen zu finden, der die geringste Summe an Kantengewichten aufweist. Übertragen auf das Belegungsgitter bildet jede Zelle einen Knoten im Graphen, welcher mit allen Knoten der Nachbarzellen verbunden ist. Deren Kantenverbindungsgewichte hängen von den Belegungswahrscheinlichkeiten ab.

A* nutzt zur effizienten Suche eine bestimmte Heuristik. Es werden bevorzugt die Pfade traversiert, die bei einer optimistischen Schätzung die kleinste prospektive Kantengewichtssumme aufweisen. Diese optimistische Heuristik nimmt bei Elfes an, dass auf einem Suchpfad sämtliche Zellen mit unbekannten Zellzustand frei sind.

Zusätzlich basieren die Kantengewichte nicht nur auf den Belegungswahrscheinlichkeiten, sondern enthalten auch einen konstanten Faktor. Somit kann der Roboter einen Pfad wählen, der nicht nur frei von Hindernissen, sondern auch möglichst kurz ist (vgl. [Elfes 1991]).

Elfes hat insgesamt viele bemerkenswerte Grundlagen entwickelt, welche heute noch vielfach in Belegungsgittern eingesetzt werden. Seine Kernidee, das Umfeld in quadratische Zellen zu diskretisieren, wird in fast

allen Arbeiten im Bereich der Belegungsgitter wieder aufgegriffen. Auf diese Umfeldrepräsentation hat er viele bekannte Algorithmen adaptiert und geeignet kombiniert. Zu nennen sind dabei die probabilistischen inversen Sensormodelle, die Bayeslogik zur Aktualisierung von belegten und freien Zellen, den A*-Algorithmus für die Pfadplanung und die Eigenbewegungs- und Kartenschätzung mittels Landmarken. Er liefert nicht nur eine theoretische Grundlage, sondern hat diese mit verschiedenen Robotern auch erfolgreich in die Praxis umgesetzt.

Insgesamt konzentriert sich Elfes mit seinen Arbeiten auf den Robotikbereich. Besonders die Erstellung von globalen Belegungskarten scheint jedoch für den Automobilbereich ungeeignet. Zum einen befahren Autos viel längere Strecken, sodass diese Karte wenn überhaupt nur äußerst speicheraufwendig umgesetzt werden kann. Zum anderen zeichnen sich die meisten Fahrten dadurch aus, dass keine Straße doppelt befahren wird, sodass man die Belegungskarte zusätzlich persistent speichern muss, um überhaupt Nutzen aus dieser zu ziehen. Auch kann man es dem Fahrer nicht zumuten, dass sein Auto stehen bleibt, nur weil die Umfeldwahrnehmung längere Zeit von Odometriefehlern ungestört Umfeldinformationen sammeln möchte.

Seine probabilistischen Modelle sind noch von vielen Vereinfachungen geprägt: Elfes nimmt beispielsweise an, dass jeder Zellzustand unabhängig von anderen ist, und behandelt widersprüchliche und unvollständige Informationen nur gemäß der Bayeslogik. Die Problematik der systematischen Fehlziele werden wenig berücksichtigt und Messungen innerhalb einer Messreihe werden als unabhängig voneinander betrachtet.

Ein positiver Aspekt ist die separate Behandlung von Sensoren. Jedem Sensor wird ein eigenes kleines Belegungsgitter zugeordnet, sodass die Möglichkeit besteht, die Messreihe dieses Sensors von den Messreihen der anderen Sensoren zu trennen. Bei der lokalen Umfeldrekonstruktion können dann andere Fusionsalgorithmen benutzt werden, sodass beispielsweise eine größere Unabhängigkeit angenommen werden kann als von Messungen, die von einem Sensor stammen.

5.5.2. Belegungsgitterfusion nach dem MURIEL-Ansatz

In [Konolige 1997] wird ein Ansatz Namens „Multiple Representation, Independent Evidence Log (MURIEL)“ vorgestellt, der versucht, bestimmte Abhängigkeiten zwischen Messungen aufzulösen. Der Ansatz ist dabei für Ultraschallsensoren optimiert, sodass in dem verwendeten inversen Sensormodell die Eigenschaften dieses Messprinzips berücksichtigt wird.

Hierbei werden insbesondere Objekte berücksichtigt, die aus bestimmten Winkeln heraus schwer zu detektieren sind. Solche Objekte haben beispielsweise planare und glatte Oberflächen, welche die Ultraschallwellen in den Raum spiegeln, aber nicht direkt zum Sensorträger zurückreflektieren. Solche Objekte können Ursache von Mehrfachreflexionen sein, was eine mehr oder weniger zufällige Positionsinformation des Objektes in der Signalauswertung ergibt.

Im Gegensatz dazu sind Objekte mit einer rauen und gewölbten Oberfläche leichter zu detektieren, da der Ultraschall diffus in alle Richtungen gestreut wird und somit auch direkt zum Sensorträger.

Um u. a. vermeintliche Freibereiche durch Mehrfachreflexionen auszuschließen, wird für jede Zelle geschätzt, ob diese diffuse oder spiegelnde Eigenschaften gegenüber einer Messung aufweist. Dieses erfolgt durch Analyse des Zellzustandes: Ist eine Zelle beispielsweise bereits von mehreren Positionen aus als belegt geschätzt worden und eine Messung verfehlt diese Zelle, dann wird eine Mehrfachreflexion mit hoher Wahrscheinlichkeit angenommen.

Die genannten Reflexionseigenschaften einer Zelle werden von einem Schätzer rekursiv angenähert und im Zellzustand abgelegt. Die Reflexionsattribute sind Teil des inversen Sensormodells, welches eine Belegungswahrscheinlichkeit dynamisch generiert. Diese wird wie bei Elfes rekursiv nach Bayes für jede Zelle aktualisiert.

In [Konolige 1997] wird angenommen, dass Belegt- und Freimessungen in einer statischen Umgebung stark korrelieren, wenn der Sensorträger die gleiche Position zu einer Zelle aufweist. Dies wird insbesondere mit konstanten Detektionseigenschaften statischer Objekte aus bestimmten Winkeln und Entfernungen begründet. So scheint es wahrscheinlich, dass wenn ein Objekt aufgrund hoher spiegelnder Eigenschaften von einer Position aus mit einer Messung nicht detektiert werden kann, dieses auch für wiederholte Messungen gilt.

Um eine Akkumulation von fehlerhaften Nichtdetektionen und falschen Positionen zu vermeiden, soll jede Zelle nur dann aktualisiert werden, wenn von der aktuellen Roboterposition nicht bereits eine Belegt- bzw. Freimessung für diese Zelle erfolgte. Insgesamt soll der Einfluss der Aufenthaltsdauer des Roboters in bestimmten Positionen auf den Zellzustand neutralisiert werden.

Um dieses zu erreichen, wird in jeder Zelle für Belegt- und Freimessungen jeweils eine Umgebungskarte verwaltet, in der die Roboterpositionen eingetragen werden, von denen die Zelle aus aktualisiert worden ist. In [Konolige 1997] wird eine diskretisierte Darstellung in Polarkoordinaten vorgeschlagen, wobei 64 Sektoren und 3 Entfernungsbereiche berücksich-

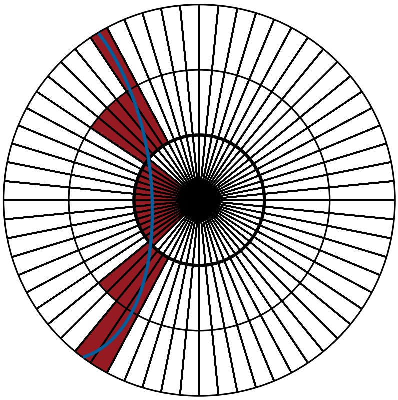


Abbildung 5.14.: Zelle mit diskretisierter Umgebungskarte in Polardarstellung. Der Sensorträger durchläuft diese (blaue Trajektorie) und misst aus bestimmten Sektorabschnitten (rote Bereiche) den Zellzustand. Dieser wird pro Sektorabschnitt nur einmal aktualisiert.

tigt werden. Die Datenstruktur wird als Bitfeld dargestellt, wobei pro Karte 192 Bits benötigt werden. Jede Zelle erhöht somit ihren Speicherbedarf für die beiden Karten um 48 Bytes. Abbildung 5.14 stellt solch eine Karte und einen möglichen Zustand exemplarisch dar.

Der vorgestellte Ansatz ist durch die Umgebungskarten in den einzelnen Zellen deutlich speicheraufwendiger als konventionelle Ansätze. Er erlaubt dafür eine bessere Auflösung von Messabhängigkeiten, welche darauf basiert, ähnlichen Messungen die gleiche Evidenzmasse zuzusprechen wie einer einzelnen Messung. Auch wird eine wohl genauere Umgebungsdarstellung dadurch erreicht, dass jede Messung mit dem Belegungsgitter plausibilisiert und entsprechend dynamisch gewichtet wird.

Zu kritisieren ist, dass der Ansatz eine vollständige Abhängigkeit[4] der Messungen aus einem Sektorabschnitt annimmt und keine partielle. So sind aber auch Effekte denkbar, die stochastisch unterschiedliche Messungen verursachen können. Sinnvoller wäre wohl eine Evidenzabnahme für wiederholte Messungen aus einem Sektorabschnitt heraus. Dies würde aber zusätzlichen Speicher kosten, da anstatt einer Markierung pro Sek-

[4] Eine Zufallsvariable y ist von einer anderen Zufallsvariable x vollständig abhängig, wenn y aus x berechnet werden kann (vgl. [Lancaster 1963]). In diesem Fall liefert y keine neue Information.

torabschnitt ein Zähler notwendig wäre.

In [Konolige 1997] wird nur die Generierung eines Belegungsgitters beschrieben, während keine Aussage über Anwendungen oder über Auswertungsalgorithmen gemacht wird. Ob mit dem MURIEL-Ansatz eine messbar bessere Performance bei spezifischen Anwendungen nach sich zieht, bleibt zu untersuchen.

5.5.3. Belegungsgitterfusion mittels erlernter inverser Sensormodelle

Die bereits vorgestellten Ansätze nutzen manuell erstellte inverse Sensormodelle, welche auf vielen Annahmen beruhen, wie beispielsweise Verteilungen von Zufallsvariablen und Gewichtungen. Durch solche Modelle kann jedoch eine komplexe Sensorcharakteristik nur mehr oder weniger vereinfacht dargestellt werden. Ein alternativer Ansatz wird in den Arbeiten von [Thrun 1998], [van Dam 1996] und [Yu 2007] verfolgt, welcher hier vorgestellt wird:

Der allgemeine Aufbau des Belegungsgitters ähnelt in diesen Arbeiten dem Ansatz von Elfes: So enthält jede Zelle genau eine Massenverteilung, die mittels inversen Sensormodellen und Logikalgorithmen aktualisiert wird. Die Zellzustände aller Zellen werden als unabhängig angenommen.

Bei der Generierung der inversen Sensormodelle verfolgen diese Arbeiten einen genaueren Ansatz als Elfes: Dieser basiert darauf, die Funktion zur Belegungsschätzung einer Zelle in einem überwachten Lernprozess zu approximieren. Hierzu nutzen alle ein mehrschichtiges künstliches neuronales Netz mit Fehlerrückführungsalgorithmus, wie es in Abschnitt A.3.5 beschrieben wird. Prinzipiell sind aber auch andere probabilistische Algorithmen aus dem Bereich des Maschinellen Lernens denkbar, wie beispielsweise die SVM (s. A.3.4) oder der kNN-Algorithmus (s. A.3.2).

Eine entscheidende Rolle beim Anlernen dieser Klassifikatoren spielen gute Lerndaten, welche repräsentativ und umfassend sein sollten. Bei der Erzeugung dieser Daten verfolgen alle drei Arbeiten unterschiedliche Methoden, welche in diesem Abschnitt vorgestellt werden.

Belegungsgitterfusion nach Thrun

Das in [Thrun 1998] beschriebene künstliche neuronale Netz soll aus Sensorsignalen und einer Zellposition die Belegungswahrscheinlichkeit dieser Zelle liefern. Im Detail besteht der Eingabevektor X aus der relativen Po-

sition der zu aktualisierenden Zelle (x,y) in Polarkoordinaten und vier synchronen Ultraschallabstandsmessungen $M = \{m_1, \ldots, m_4\}$:

$$X = \left(m_1, m_2, m_3, m_4, \sqrt{x^2 + y^2}, \arctan\left(\frac{y}{x}\right)\right) \tag{5.32}$$

Der Ausgabevektor besteht aus einer einzelnen Belegungswahrscheinlichkeit für die zum Eingabevektor korrespondierenden Zelle c:

$$Y = P(H_B|c) \tag{5.33}$$

Um das künstliche neuronale Netz anzulernen, werden in [Thrun 1998] Lerndaten aus synthetischen idealen Karten durch manuell erstellte Sensormodelle simulativ generiert. Auf solch einer Karte wird der Sensorträger zufällig positioniert und ausgerichtet. Aus den Kartendaten wird anschließend eine mögliche Messung gemäß dem stochastischen Modell für den jeweiligen Sensor erzeugt. Die Ground Truth-Belegungsinformation für den Ausgabevektor lässt sich aus der Karte entnehmen.

Zur Generierung weiterer Lerndaten werden weitere Positionen und Ausrichtungen für den Sensorträger zufällig generiert und wie beschrieben ausgewertet. Hierbei wird eine Gleichverteilung der Zufallsvariable auf möglichen Kartenpositionen angenommen, was exemplarisch in Abbildung 5.15 dargestellt ist.

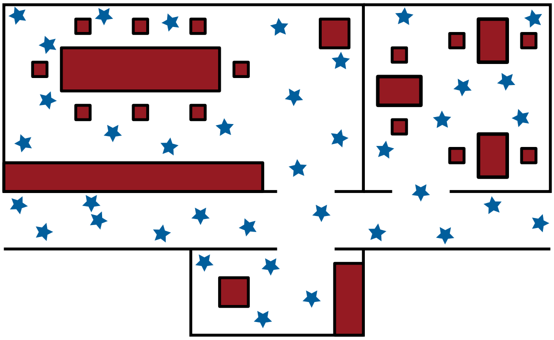


Abbildung 5.15.: Synthetische Karte mit gleichverteilten Roboterpositionen und -ausrichtungen (blaue Sternchen)

Der Ansatz nach [Thrun 1998] erzeugt unter bestimmten Umständen ein genaueres inverses Sensormodell als eine manuelle Modellierung. Er basiert jedoch auch auf ähnlich vielen Modellannahmen:

Sowohl die repräsentative Modellierung der Karte ist herausfordernd als auch die genaue Darstellung des Sensormodells. Für Ersteres müssen beispielsweise innerhalb von Gebäuden die Türen, die Einrichtungsgegenstände und die Räumlichkeiten möglichst realitätsnah und entsprechend häufig dargestellt werden, während beim Sensormodell für eine hohe Genauigkeit sämtliche Messphänomene berücksichtigt werden sollten, die in der Domäne auftreten können. Hierzu zählen neben den obligatorischen Messungenauigkeiten auch Doppeldeutigkeiten durch Mehrfachreflexionen oder Absorptionen, welche beispielsweise durch Material und Ausrichtung von Objektoberflächen verursacht werden können. Außerdem können Unterschiede in der Signalcharakteristik auftreten, welche durch den Dynamikzustand des Sensorträgers, durch Interferenzen mit anderen Sensorsignalen oder mit der Verletzung der statischen Umgebungsannahme durch dynamische Objekte oder temporären Verdeckungen begründet werden können.

Der Zustandsraum der Lerndaten zeigt, dass zwar zeitgleiche Messungen als abhängig angenommen werden, aber zeitlich auseinanderliegende Messungen als unabhängig. Persistente Scheinziele oder persistent nicht detektierbare Objekte werden somit nicht berücksichtigt.

Trotz dieser Schwächen ist die Vorgehensweise in [Thrun 1998] attraktiv, wenn das Sensormodell deutlich leichter zu generieren ist als dessen inverses Pendant. Dies ist bei den verwendeten Ultraschallsensoren der Fall, welche ausschließlich eine radiale Entfernungsmessung liefern, aus denen jedoch keine eindeutige Objektposition und -kontur ermittelt werden kann. Umgekehrt ist die Generierung einer beispielsweise normalverteilten Abstandsmessung zu einer gegebenen Karte recht einfach möglich. Des Weiteren erlaubt der simulative Ansatz die Generierung einer großen Lerndatenmenge mit geringem Mehraufwand.

Viele andere Sensoren, wie Laserscanner, Stereokamerasensoren oder auch einige Radarsensoren schätzen bereits in der Signalvorverarbeitung eine mehr oder weniger genaue Objektposition und Objektkontur. Für diese Sensoren ist die manuelle Generierung eines Sensormodells nicht viel komplexer zu sein als dessen inverses Pendant. Eine höhere Genauigkeit für solche inversen Sensormodelle ist wohl nur mit realen Lerndaten zu erwarten.

Die in [Thrun 1998] entwickelten Ansätze sind auf verschiedene Roboter portiert worden, sodass die generierten Belegungsgitter auch praktisch genutzt werden. Eine dieser Anwendungen ist der Museumsführer „Rhino“ [Burgard 1998], der Besucher von Exponat zu Exponat leitet und Hintergrundinformationen zu diesen liefert. Er nutzt dazu die erlernten inversen Sensormodelle, um eine initiale bekannte Karte des Museums zu erweitern

und auf dieser Karte eine möglichst sichere Trajektorie zum nächsten Exponat zu finden. Diese dynamische Erweiterung ist für neue Hindernisse, wie stehende Besucher und verschobene Gegenstände sinnvoll.

Der Pfadplaner basiert wie bei Elfes im Wesentlichen auf dem A*-Algorithmus, wurde jedoch um zwei Aspekte erweitert. Zum einen wird der berechnete Pfad noch eventuell von einer Routine modifiziert, die zur Kollisionsvermeidung von dynamischen Hindernissen wie Besuchern dient, die noch keine ausreichende Evidenz im Belegungsgitter verursacht haben. Eine andere Erweiterung erhöht die Echtzeitfähigkeit, indem jederzeit eine geeignete Trajektorie mit geringer Latenz abgefragt werden kann, auch wenn diese aktuell nicht der optimalen Trajektorie nach der A*-Kostenfunktion entspricht.

Insgesamt konnte der Belegungsgitteransatz nach [Thrun 1998] gute Ergebnisse in der gewählten Domäne erzielen, da er eine recht hohe Verfügbarkeit und Funktionalität erreicht hat. Es zeigte sich auch ein großes Besucherinteresse am maschinellen Museumsführer, der nicht nur andere Exponate erklären konnte, sondern auch selbst ein solches darstellte.

Einschränken muss man aber, dass die Domäne nur aus einem Teil eines Museums bestand und somit im Vergleich zu typischen Automotivedomänen sehr klein ist. Außerdem wird eine virtuelle Belegungskarte vorgegeben, die nur noch dynamisch erweitert wird. Eine beliebig große Auswahl an Objekten kann dort bereits abgelegt werden, sodass insbesondere schwierig zu detektierende Gegenstände mit dieser Methode problemlos behandelt werden können. Diese Möglichkeit ist in Automotivedomänen praktisch nicht gegeben.

Belegungsgitterfusion nach Yu

Die in [Yu 2007] verwendete Netzfunktion ähnelt der von [Thrun 2006]: Es werden sowohl Zellposition als auch Messungen als Eingangsvektoren verwendet und eine einzelne Zellzustandsschätzung wird in Form einer Massenverteilung als Ausgabevektor generiert. Auch wird der Ausgabevektor als unabhängig angenommen und mittels Logikalgorithmen mit dem Zellzustand verknüpft. Als Unterschied ist zu vermerken, dass in den Eingabevektor zusätzlich die zwei nächstälteren Messungen einfließen und dass eine Massefunktion nach der Dempster-Shafer Theorie und nicht nach Bayes im Ausgabevektor Verwendung findet.

Insgesamt erlaubt der Ansatz die Abhängigkeiten von drei aufeinanderfolgenden Messungen zu erlernen. Nachteilig bezüglich der statistischen Genauigkeit ist jedoch, dass eine einzelne Messung dreimal in den Eingabevektor der Netzfunktion einfließt und die Netzausgaben dennoch als un-

abhängig voneinander betrachtet werden. Somit kann das künstliche neuronale Netz zum Teil einzelne Messausreißer neutralisieren, jedoch liefert es wohl keine Möglichkeit, persistente Abhängigkeiten im finalen Zellzustand abzubilden.

Um Lerndaten für das künstliche neuronale Netz zu generieren, wird in [Yu 2007] ein halbautomatischer Ansatz verfolgt. Die Ausgabevektoren werden wie in [Thrun 1998] aus einer Ground Truth-Karte generiert: Um diese zu erzeugen, wird der Roboter in einem Raum positioniert und alle Objekte, Wände und sonstige in Sensorreichweite befindliche Hindernisse werden manuell ausgemessen. Die Ergebnisse werden in die Ground Truth-Karte übertragen, welche zusammen mit den Messdaten die Lernbasis bildet.

Insgesamt erfordert diese Methode eine Umgebung mit klaren und scharfen Konturen, um eine Ground Truth-Karte eindeutig generieren zu können. Um eine Überanpassung an wenige Lerndaten zu vermeiden, ist es wohl auch notwendig, mehrere Räume in die Lernbasis mit einfließen zu lassen. Kleinere Ungenauigkeiten könnten sich aus der manuellen Vermessung der Räumlichkeiten ergeben. Dennoch ist diese aufwändige Methode der simulativen Generierung vorzuziehen, da die Lerndaten deutlich realistischer sind, als die synthetisch generierten.

Für den Automotivebereich ist diese Methode wohl zu aufwendig, da moderne Sensoren eine viel größere Reichweite haben als die im Robotikbereich verbreiteten Ultraschallsensoren und somit eine deutlich größere Ground Truth-Karte erforderlich wäre. Fehlende Wandbegrenzungen vergrößern den Raum zusätzlich. Auch sind Begrenzungen als solche zum Teil schwierig zu definieren, wie beispielsweise bei Rasenflächen mit unterschiedlich hohem Bewuchs oder bei kleinen aufgeschütteten Schneehaufen.

In [Yu 2007] ist nur die Erzeugung der Belegungsgitter beschrieben worden, jedoch keine spezifische Anwendung. Auf eine Bewertung dieser Methode bezüglich einer konkreten Problemstellung wird deshalb verzichtet.

Belegungsgitterfusion nach van Dam

Ein weiterer Ansatz zur Erstellung eines inversen Sensormodells ist in [van Dam 1996] zu finden. Im Gegensatz zu den bisher vorgestellten Ansätzen kommt die dort beschriebene Netzfunktion mit einer einzelnen Messung ohne Zellpositionsangabe als Eingabevektor aus.

Der Ausgabevektor besteht dafür aus einem kompletten lokalen Belegungsgitter, aus dem man die Belegungswahrscheinlichkeit einer ge-

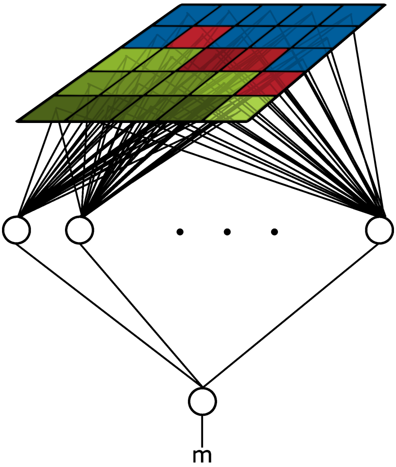


Abbildung 5.16.: In [van Dam 1996] verwendetes künstliches neuronales Netz zur Generierung eines inversen Sensormodells

wünschten Zelle herausextrahieren kann. Dieses Netz ist mit einer beispielhaften Messung m in Abbildung 5.16 dargestellt.

Vorteilhaft an diesem Ansatz ist, dass die Netzfunktion des künstliche neuronale Netz für jede Messung nur einmal ausgeführt werden muss und nicht für jede einzelne Zelle, wie bei den anderen Ansätzen.

Nachteilig ist, dass aufgrund des großen Ausgabevektors eine wohl größere Menge an Lerndaten notwendig ist, als bei vergleichbaren Ansätzen. Auch die fehlende Positionsangabe im Eingabevektor begründet dies, da die Abhängigkeiten und Ähnlichkeiten von Nachbarzellen umständlich aus den Ground Truth-Daten abgeleitet werden muss. Eine Interpolation mit Hilfe von Zellabstand und -winkel ist somit nicht so einfach möglich wie bei anderen Ansätzen.

Wie auch der Ansatz von [Thrun 1998] erlaubt dieser keine Erlernung von Abhängigkeiten zwischen Messungen. Vielmehr mittelt das künstliche neuronale Netz diese über die Gesamtheit der Lerndaten.

In [van Dam 1996] werden keine synthetischen sondern reale Lerndaten erzeugt: Dazu wird der Ansatz einer prospektiven Messdatenanalyse, ähnlich wie in Abschnitt 4.6.2 beschrieben, mit einer Referenzsensorik gekoppelt. Hierzu befährt der Roboter eine Zufallstrajektorie bis zur Kollision mit einem Objekt. Letztere kann beispielsweise durch Berührungssensoren detektiert werden, welche die Referenzsensorik darstellt. Die befahre-

ne Trajektorie wird als sicher frei deklariert, während die Zelle, in der die Kollision stattfand, als belegt angesehen wird. Diese Informationen bilden zusammen mit den Messungen, die während der Fahrt aufgenommen worden sind, die Lerndaten.

Diese Vorgehensweise ist bei Robotern bei geringen Geschwindigkeiten Erfolg versprechend, jedoch für Versuchsträger im Automotivebereich aufgrund des hohen Schadenpotenzials nicht akzeptabel.

Auch in [van Dam 1996] ist nur die Belegungsgittergenerierung ohne spezifische Anwendung beschrieben worden, sodass die Bewertung des Ansatzes ohne Problemstellung verzichtet wird.

5.5.4. Bewertung

Die vorgestellten Ansätze aus der Robotik bemühen sich größtenteils um eine genaue Darstellung der inversen Sensormodelle. Allerdings gelingt es diesen nur teilweise die Abhängigkeiten aufzulösen: Während bei Elfes unterschiedliche Sensoren separat betrachtet werden können, wird beim Muriel-Ansatz eine Akkumulation von ähnlichen Messungen verhindert. Bei Yu können hingegen einzelne Messausreißer kompensiert werden.

Was allen Ansätzen fehlt, ist eine Abhängigkeitsbeschreibung der Zellzustände. So können beispielsweise mehrere Zellen durch eine einzelne Fehlmessung fälschlicherweise als belegt oder frei angesehen werden: Dass die Zustände benachbarter Zellen deshalb stark miteinander korrelieren können, wird durch keinen Ansatz dargestellt. Die Konzentration der vorgestellten Arbeiten beschränkt sich darauf, die Abbildungsoperation von Messungen auf Einzelzellzustände zu perfektionieren, ohne dass das Belegungsgitter in der Gesamtheit betrachtet wird.

Nur in einem Ansatz wird eine Auswertung des Belegungsgitters beschrieben: Zur Pfadfindung wird bei Thrun eine Variante des A*-Algorithmus eingesetzt, welcher die Zellzustände als unabhängig betrachtet und eine Kostenfunktion dementsprechend minimiert.

Bei vielen vorgestellten Ansätzen ist eine Übertragung auf den Automotivebereich problematisch, da die natürliche Domäne eines Autos sehr viel größer als die eines Roboters ist. Die Verwaltung eines globalen Belegungsgitters würde sehr große Mengen an Speicher kosten und es könnte wohl auch nur nach mehrmaligem Befahren einer Strecke sinnvoll genutzt werden. Auch erschwert die autotypische Domäne die Ground Truth-Datengenerierung der gesamten Umgebung aufgrund großer Sensorreichweiten, hoher Geschwindigkeiten und dynamischer Objekte.

Im Folgenden soll deshalb nur das unmittelbare Umfeld des Autos in einem Belegungsgitter abgelegt werden; Sensorreichweiten und Applikation implizieren hierbei die Größe des Belegungsgitters. Typische Probleme aus der Robotik entfallen damit aus der Betrachtung: Einerseits ist eine Lokalisation in einem globalen Belegungsgitter nicht notwendig und andererseits benötigt man keine Algorithmen, die komplexe Pfade in einem sehr großen Belegungsgitter finden müssen.

5.6. Abhängigkeiten in Belegungsgittern

Die im Abschnitt 5.2 gesammelten Anforderungen verlangen unter anderem eine genaue und statistisch korrekte Schätzung unter realistischen Umgebungen und Anwendungen. Hierzu gehört die Erkennung und Neutralisierung von Abhängigkeiten zwischen Zellen und Messungen.

Bevor entsprechende Algorithmen entwickelt werden, werden diese beiden klassischen Probleme konventioneller Fusionsalgorithmen vorgestellt und formal definiert. Diese treten einerseits bei der Fusionierung mehrerer Zellen zu einem Gesamtzustand auf und andererseits dann, wenn Messungen nicht unabhängig voneinander sind. Diese Phänomene sind praktisch bei jeder Belegungsgitteranwendung und in jeder Messreihe eines Sensors allgegenwärtig.

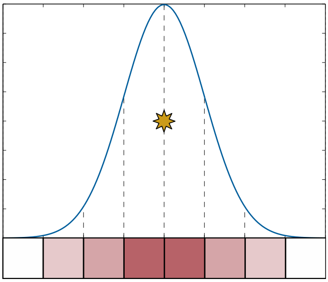
5.6.1. Inter-Cell Dependency Problem

Häufig verwendete inverse Sensormodelle, die eine unsichere Position im Rahmen einer Messung darstellen, basieren auf bestimmten Verteilungsannahmen, wie beispielsweise einer Normalverteilung (vgl. [Elfes 1989a]). Diese Wahrscheinlichkeitsdichtefunktion wird über das Belegungsgitter projiziert und die entsprechenden Zellen werden mit der überlappenden Fläche oder dem Funktionswert aktualisiert. Aus Performancegründen werden meist nur solche Zellen aktualisiert, welche sich signifikant mit dieser Verteilung überlappen.

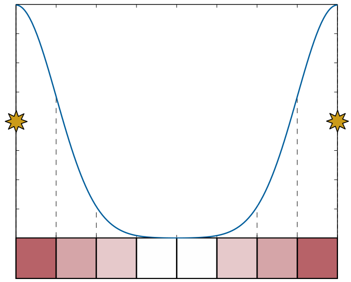
Diese Algorithmen können eine statistisch genaue Schätzung der Belegungswahrscheinlichkeit einer einzelnen Zelle im Gitter leisten. Viele Anwendungen benötigen jedoch eine akkumulierte Schätzung über eine Menge von Zellen und nicht eine Menge von Schätzungen einzelner Zellen. Beispielhaft seien Pfadfindungsalgorithmen genannt, welche verschiedene Pfadalternativen bewerten müssen. Um die Pfade zu vergleichen, muss die jeweilige gesamte Zellmenge betrachtet werden, die von diesem Pfad

überdeckt werden. Für diese Anforderung muss ein Abfragealgorithmus entwickelt werden, der eine Menge von Zellen zu einem Zustand fusioniert.

Unter klassischer Benutzung bekannter Logikalgorithmen wie Bayes, Dempster-Shafer oder Dezert-Smarandache Theorie ist die Entwicklung von Abfragealgorithmen, die solch eine statistisch genaue Belegungsschätzung liefern sollen, nicht möglich. Im Allgemeinen ist es unentscheidbar, ob eine gegebene Schätzung den Zustand einer Zellmenge statistisch korrekt repräsentiert, wenn jede Zelle nur die Informationen über den eigenen Zustand enthält.



(a) Erzeugte Belegtevidenz für einzelne Zellen (rote Quadrate) verursacht von einer einzelnen Messung (gelber Stern), die mitten im Fahrschlauch liegt



(b) Erzeugte Belegtevidenz für einzelne Zellen (rote Quadrate) verursacht von zwei Messungen (gelbe Sterne), die genau auf den Rändern des Fahrschlauches liegen

Abbildung 5.17.: Zwei unterschiedliche Messmengen (gelbe Sternchen) erzeugen zwei gleiche Zellmultimengen (rote Kästchen), aber implizieren unterschiedliche Befahrbarkeitswahrscheinlichkeiten

Ein Beispiel dazu wird in Abbildung 5.17 gezeigt, wo zwei äquivalente Zellmengen konstruiert werden. Diese könnten beispielsweise einen eindimensionalen Fahrschlauch darstellen, in der die Anwendung eine Befahrbarkeitswahrscheinlichkeit bestimmen möchte.

Hierbei zeigt die Abbildung 5.17a eine punktförmige Belegtmessung, die komplett innerhalb des Fahrschlauches liegt und somit eine Befahrbarkeitswahrscheinlichkeit von null erzeugt[5]. Die Abbildung 5.17b hingegen

[5] Jede Normalverteilung überdeckt einen unendlich großen Raum. Zur Vereinfachung werden Zellen, die sehr weit weg von der Messung liegen nicht berücksichtigt. Für sie wird approximativ eine neutrale Belegungswahrscheinlichkeit von null angenommen.

repräsentiert zwei punktförmige Belegtmessungen, deren Erwartungswerte bzgl. Position genau auf den beiden Rändern der Fahrschlauchgrenzen liegen. Somit ergibt sich eine Chance von $1/2$ für jede punktförmige Messung, dass sich dieses außerhalb des Fahrschlauches befindet. Unter der Annahme, dass die beiden Messungen unabhängig sind, ergibt sich damit eine Wahrscheinlichkeit von $1/4$, dass beide Messungen außerhalb des Fahrschlauches liegen und dieser somit befahrbar ist. Da jedoch beide Zellmultimengen äquivalent sind, existiert keine Funktion, die diese beiden Fälle unterscheiden könnte.

Abhängig davon, ob die Belegungsevidenz einer Zellmenge von einer oder mehreren Messungen stammt, benötigt man unterschiedliche Operatoren, um die Schätzung genau zu gestalten. Sind zwei Evidenzen e_1 und e_2 in dieser Form stochastisch abhängig, so muss man die verteilte Evidenz der WDF mit Hilfe des Additionsoperators zusammenfassen:

$$P_{H_B}(e_1, e_2) = e_1 + e_2 \tag{5.34}$$

Bei stochastisch unabhängigen Evidenzen e_1 und e_2 ergibt sich die Belegungswahrscheinlichkeit P_{H_B} wie folgt:

$$P_{H_B}(e_1, e_2) = e_1 + e_2 - e_1 \cdot e_2 \tag{5.35}$$

Abhängig von der Anzahl der Messungen und deren Konstellation können große Unterschiede zwischen geschätzten und wahren Belegungswahrscheinlichkeiten auftreten, je nachdem wie fehlerhaft die Abhängigkeitsannahme ist. Um den maximalen Fehler zu quantifizieren, wird im Folgenden ein weiteres Belegungsgitter künstlich erzeugt.

Es sei angenommen, dass n nichtüberlappende Messungen innerhalb eines Belegungsgitters so fusioniert werden, dass jede Einzelne von ihnen die Zellmenge nur partiell überdeckt, die vom Abfragealgorithmus genutzt wird. Jede Messung überlappt diese mit einer Fläche von $\frac{1}{n}$, sodass die Summe der Belegungsevidenzen eins ergibt. Unter der Annahme, dass jede Messung unabhängig von den anderen Messungen ist, beträgt die Belegtwahrscheinlichkeit $P_{H_B}(n)$:

$$P_{H_B}(n) = 1 - \left(1 - \frac{1}{n}\right)^n \tag{5.36}$$

Wenn die Anzahl der Messungen erhöht wird, konvergiert die Freiwahrscheinlichkeit $P_{H_B}(n)$ zu einer Konstante:

$$\lim_{n \to \infty} P_{H_B}(n) = 1 - \frac{1}{e} \tag{5.37}$$

Als Ergebnis erhält man eine Belegungswahrscheinlichkeit von bis zu 0,64 abhängig von der Anzahl der Messungen. Unter der gegensätzlichen Annahme, dass sämtliche Belegungswerte aus einer einzelnen Messung stammen und somit abhängig sind, würde man eine Belegungswahrscheinlichkeit von 1,0 erhalten. Diese Beispiele führen zu folgendem Theorem:

Theorem 1 (Unentscheidbarkeit der Befahrbarkeit bei klassischen Belegungsgitteransätzen)**.**
Gegeben sei eine Menge von Messungen, welche die Zellen und Zellzustände eines Belegungsgitters mit klassischen Logikfunktionen und inversen Sensormodellen aktualisiert. Ferner sei aus den Messungen entscheidbar, ob das gesamte Belegungsgitter überfahrbar ist. Dann existiert keine Funktion, welche die Befahrbarkeit des Belegungsgitters anhand der Zellzustände im Allgemeinen entscheiden kann.

Der Grund für die Schätzunterschiede ist der Informationsverlust innerhalb des Fusionsprozesses: Der Grad der Abhängigkeit eines Zellzustandes zu anderen Zellen ist nicht in der Belegungswahrscheinlichkeit einer einzelnen Zelle dokumentiert, welches das „Inter-Cell Dependency Problem (ICDP)“ verursacht. Eine sorgfältige Entwicklung von konsistenten Fusions- und Abfragealgorithmen ist notwendig, um dieses Problem zu lösen.

Für die Entwicklung solcher Algorithmen ist die formale Definition dieses Problems hilfreich. Hierbei müssen sowohl Fusions- als auch Abfragealgorithmus so aufeinander abgestimmt werden, dass diese bestimmte Eigenschaften aufweisen. Dazu sei folgende Definition aufgestellt:

Definition 4 (Inter-Cell Dependency Problem)**.**
Gegeben seien zwei Mengen von Zellen C und D, welche den gleichen Raum mit unterschiedlichen Zellgrößen bedecken. C und D werden mit der gleichen Menge von Messungen M durch einen Fusionsalgorithmus $\xi(C,M)$ bzw. $\xi(D,M)$ aktualisiert.

Ein Abfragealgorithmus $\beta(X)$ löst das Inter-Cell-Dependency Problem, genau dann wenn $\beta(\xi(C,M)) = \beta(\xi(D,M))$ für alle möglichen Mengen von C, D und M ist.

Die Definition beinhaltet insbesondere den Fall, dass das Ergebnis des Abfragealgorithmus einer Menge von Zellen C genau dem Ergebnis entsprechen muss, als wenn eine einzelne große Zelle $D = \{d\}$ den gleichen Bereich überdeckt. Bei der Diskretisierungsgenauigkeit verlangt die Definition keine Begrenzung, sodass die Menge der Zellen C theoretisch aus unendlich vielen kleinen Zellen bestehen kann. Insgesamt sind Algorithmen, die das ICDP lösen, invariant gegenüber Zellgrößenveränderungen,

solange die Diskretisierungsgrenzen genau den Rand des Abfragebereiches erreichen.

5.6.2. Inter-Measurement Dependency Problem

Neben Abhängigkeiten von Zellzuständen existieren auch Abhängigkeiten zwischen Messungen einer Messreihe, welche beispielsweise durch persistente Scheinziele verursacht werden können. Auch können bei verwandten Messprinzipien Abhängigkeiten zwischen Messungen verschiedener Sensoren auftreten. Analog zu ICDP wird dieses Problem im Folgenden „Inter-Measurement Dependency Problem (IMDP)“ genannt.

Die Fusionsalgorithmen Bayes, Dempster-Shafer und Dezert-Smarandache Theorie sind unter klassischer Verwendung nicht in der Lage, die Abhängigkeiten in Messreihen in einem Belegungsgitter aufzulösen. Der Grund dafür ist, dass der Typ der Informationsquelle im Zellzustand nicht dokumentiert wird. So ist es schwer oder gar unmöglich zu entscheiden, aufgrund welcher Informationsquellen der Zellzustand entstanden ist. Diese sind aber notwendig, um die Abhängigkeit neuer Messdaten mit dem Zellzustand bestimmen zu können.

So ist beispielsweise eine Zelle sehr wahrscheinlich belegt, wenn diese durch drei Messungen von technologisch heterogenen Sensoren als solche bestätigt wird. Eine nicht so hohe Belegungswahrscheinlichkeit ist anzunehmen, wenn ein einzelner Sensor dreimal hintereinander eine Zelle als belegt misst. Dies ist mit persistenten Fehldetektionen und fehlerhaften Detektionen aufgrund Umgebungsartefakten für einzelne Sensoren begründet (vgl. Kapitel 2). Insgesamt ist anzunehmen, dass die Evidenzmasse für wiederholte Messungen unter ähnlichen Messumständen deutlich sinkt, wenn man den Zellzustand berücksichtigt. Wenn beispielsweise ein Sensor eine Zelle bereits vielfach als belegt postuliert hat, so scheint es kaum ein Mehrwert zu sein, wenn der Sensor dieses ein weiteres Mal tut.

Eine besondere Bedeutung erhält die Abhängigkeitsanalyse, wenn widersprüchliche Messungen für eine Zelle vorliegen. Je nach Gewichtung kann der eine oder andere Sensor die Entscheidungshoheit über den Zellzustand gewinnen.

Wie beim ICDP ist eine formale Definition für Anforderungen an einen Algorithmus sinnvoll. Im Unterschied zum ICDP werden die Zellen in einem Belegungsgitter nur einzeln betrachtet:

Definition 5 (Inter-Measurement Dependency Problem)**.**
Ein Abfragealgorithmus $\beta(\{c\})$ *löst das Inter-Measurement Dependency*

Problem, wenn er eine statistisch genaue und genaue Schätzung für alle einzelnen Zellen innerhalb eines Belegungsgitters $c \in C$ liefert.

Zu bemerken ist, dass in der Definition keine Angaben gemacht werden, was „genau“ exakt bedeutet. Dies ist insbesondere bei Massenverteilungen mit vielen Attributen wie beispielsweise bei der Dezert-Smarandache Theorie nicht einfach zu beantworten. Ob die notwendige Genauigkeit erreicht worden ist, kann zudem nur anhand einer konkreten Fahrerassistenzapplikation bewertet werden. Insgesamt bleibt die Definition damit absichtlich etwas unscharf.

Die Genauigkeitsanforderung ist jedoch wichtig, um triviale Massefunktionen auszuschließen. So scheint eine Massefunktion im Sinne der Definition zwar statistisch korrekt zu sein, wenn diese konstant maximales Unwissen oder A-priori-Wahrscheinlichkeiten als Ergebnis liefert, jedoch zu ungenaue für praktisch alle Anwendungen.

5.6.3. Kombination von ICDP und IMDP

Fusions- und Abfragealgorithmen, die die ICDP-Definition erfüllen, erlauben zwar die Kombination von Zellen zu einem Ergebnis, welches die Abhängigkeiten der Zellzustände auflösen und den Einfluss der Zellgrößen im Rahmen der Diskretisierungsgrenzen neutralisieren. Jedoch werden an das Abfrageergebnis überhaupt keine qualitativen Anforderungen gestellt: So wird die ICDP-Definition beispielsweise von einer Funktion erfüllt, die unabhängig von der Zellmenge eine konstante Verteilung als Ergebnis zurückliefert. Dieses Ergebnis ist jedoch für alle Fahrerassistenzsystemapplikation praktisch unbrauchbar.

Algorithmen, die die IMDP-Definition erfüllen, erlauben zwar eine genaue, robuste und statistisch genaue Befahrbarkeitsschätzung im Sinne der Fahrerassistenzsystemapplikation, jedoch nur für eine einzelne Zelle und nicht für eine Menge von Zellen. Auch hier ist eine Praxisrelevanz nicht gegeben, da das Ego-Fahrzeug praktisch immer mehrere Zellen überfährt.

Insgesamt benötigt die Fahrerassistenzsystemapplikation im Regelfall einen Abfragealgorithmus, der sowohl die Zellabhängigkeiten einer beliebigen Menge an Zellen berücksichtigt, als auch eine gute Schätzung im Sinne ihrer Anforderungen für diese Zellmenge berechnet. Ein Algorithmus erfüllt diese Anforderung, wenn dieser sowohl die ICDP- als auch IMDP-Definition erfüllt. Hierzu sei folgendes Theorem aufgestellt:

Theorem 2 (Kombination ICDP und IMDP)**.**
Ein Abfragealgorithmus $\beta(X)$ liefert eine genaue, robuste und statistisch genaue Schätzung im Sinne einer Fahrerassistenzsystemapplikation für

eine Menge von Zellen C, wenn er sowohl das ICDP als auch das IMDP löst.

Das Theorem folgt direkt aus der Definition von ICDP und IMDP. Sei C eine Menge von Zellen und $\{d\}$ eine Menge bestehend aus einer einzelnen Zelle d, welche die gleichen Bereiche abdecken. Beide Mengen werden mit einer Messreihe M mittels Fusionsalgorithmus $\xi(C,M)$ bzw. $\xi(\{d\},M)$ aktualisiert. Aus der ICDP-Definition folgt:

$$\beta(\xi(C,M)) = \beta(\xi(\{d\},M)) \tag{5.38}$$

Da der Abfragealgorithmus für die einzelne Zelle d nach der IMDP-Definition statistisch korrekt ist, muss dieses auch für das äquivalente Ergebnis $\beta(\xi(C,M))$ gelten. Dieser Zusammenhang ist in Abbildung 5.18 dargestellt.

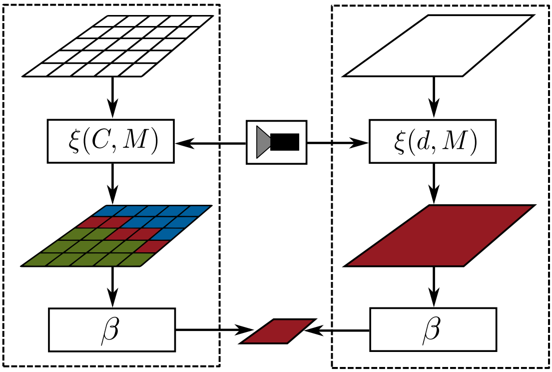


Abbildung 5.18.: Fusions- und Abfragealgorithmus, der das ICDP exemplarisch löst

Die ICDP- und IMDP-Definition wurde in den vorherigen Schritten separat definiert, sodass die Möglichkeit besteht, beide Aspekte separat behandeln zu können. So ist es beispielsweise möglich, einen ICDP-konformen Algorithmus zu entwickeln und dann für unterschiedliche Applikationen verschiedene Varianten von IMDP-Algorithmen auf das Ergebnis darauf anzuwenden. Genau diese getrennte Betrachtungsweise wird in den nachfolgenden Abschnitten angewendet, in denen ICDP- und IMDP-kompatible Algorithmen entwickelt werden.

5.6.4. Zusammenfassung

In diesem Abschnitt wurden die allgemeinen Anforderungen einer Sensordatenfusion an das eingebettete Belegungsgitter konkretisiert und dabei die Aspekte der Genauigkeit und Robustheit genauer untersucht. Die beiden zugrunde liegenden Teilprobleme ICDP und IMDP wurden dargestellt und formal definiert. Es wurde in einem Theorem bewiesen, dass einige Anforderungen automatisch erfüllt werden, wenn entsprechende Belegungsgitteralgorithmen sowohl das ICDP als auch das IMDP lösen.

Die Anforderungen zeigen auch, dass es nicht ausreicht, ausschließlich Fusionsalgorithmen für einzelne Zellzustände zu entwickeln. Ausschlaggebend ist die Kombination von Abfrage- und Fusionsalgorithmus. Die beiden Teilkomponenten einer Belegungsgitterfusion müssen genau aufeinander abgestimmt werden, um ein genaues Ergebnis zu erhalten.

Solch ein kombinierter Ansatz wird von keinem der im Abschnitt 5.5 vorgestellten Fusionsalgorithmen verfolgt. Es ist außerdem für diese Fusionsalgorithmen unmöglich, einen ICDP-fähigen Abfragealgorithmus mit hoher Genauigkeit zu entwickeln, da die Abhängigkeitsinformationen zwischen den Zellzuständen unwiderruflich verloren gegangen sind. Somit können diese Algorithmen die genannten Anforderungen nicht erfüllen.

Einige der vorgestellten Ansätze versuchen zumindest die Messabhängigkeiten nach bestimmten Aspekten aufzulösen, scheitern jedoch spätestens bei der Akkumulation des Zellzustandes durch Verwendung klassischer Logikalgorithmen. Diese nehmen eine vollständige Unabhängigkeit der Massenverteilungen von Zellzustand und neuer Messung an, was Zweifel an der statistischen Korrektheit und Genauigkeit der Schätzung aufkommen lässt, da aufeinanderfolgende Messungen praktisch immer partiell abhängig sind. Je nach Anwendung kann dies Probleme mit der Genauigkeitsanforderung verursachen.

Da die vorgestellten Ansätze aus der Literatur diese Abhängigkeiten nicht berücksichtigen, ist das Ziel der nächsten Abschnitte, geeignetere Algorithmen zu entwickeln. Diese sollen die Genauigkeit und Robustheit der Umgebungsschätzung mittels Belegungsgitter steigern.

5.7. Entwicklung von ICDP-Algorithmen

In diesem Abschnitt werden zwei Algorithmen vorgestellt, welche das ICDP zumindest approximativ lösen. Der eine ist ein einfacher nichtprobabilistischer Algorithmus, während der andere verschiedene probabilistische As-

pekte integriert. Hierzu gehört eine Vertrauensmasse pro Messung und die Möglichkeit, Messungenauigkeiten mit zu modellieren.

Beide Algorithmen basieren auf der die Dezert-Smarandache Theorie mit dem vereinfachten Wahrnehmungsrahmen Θ aus Abschnitt 5.4.1, sodass diese eine Massefunktion über vier Attribute zurückliefern. Die Verwendung anderer Logiken ist jedoch transparent möglich.

5.7.1. Nichtprobabilistischer Algorithmus

Der nichtprobabilistische Fusionsalgorithmus unterscheidet vier diskrete Zustände einer Zelle nach der Dezert-Smarandache Theorie: Entweder ist deren Zustand belegt, frei, unbekannt oder konfliktbehaftet. Dieses Modell wird auch beim Abfragealgorithmus $\beta(X)$ und bei den Messungen angewendet, wobei bei Letzterem der konfliktbehaftete Zustand entfällt. Der Algorithmus berücksichtigt weder Messungenauigkeiten noch Messunsicherheiten in Form von Scheinzielen oder fehlerhaften Nichtdetektionen.

Der Zustand einer Zelle bestimmt sich aus der Menge der Messungen, die die Zelle geometrisch betrifft. Für die einzelnen Messungen wird angenommen, dass mittels inversen Sensormodells eine geometrische Form generiert wird. Diese Form beschreibt Frei- oder Belegtbereiche. Die Menge der Messungen bestimmt, welche Ereignisse des Wahrnehmungsrahmens Θ für eine Zelle eingetreten ist. Die Eintrittsbedingungen werden wie folgt definiert:

$$P(H_F|M,c) = \begin{cases} 1, & \text{wenn die geometrische Vereinigung aller Frei-} \\ & \text{messungen die gesamte Zelle als frei bestätigt} \\ 0, & \text{sonst} \end{cases} \tag{5.39}$$

$$P(H_B|M,c) = \begin{cases} 1, & \text{wenn eine Messung einen Teil} \\ & \text{der Zelle als belegt bestätigt} \\ 0, & \text{sonst} \end{cases} \tag{5.40}$$

Ob H_F eingetreten ist, hängt allein von der Menge der Messungen ab, die eine Freihypothese stützen. H_B hingegen tritt nur aufgrund einer oder mehrerer Messungen ein, die eine Belegthypothese stützen. Der Zellzustand einer Zelle c ergibt sich aus den Kombinationsmöglichkeiten der eingetretenen Ereignisse des Wahrnehmungsrahmens Θ:

$$\xi(c,M) = \begin{cases} H_F & \text{, wenn } (P(H_F|M,c) = 1) \wedge (P(H_B|M,c) = 0) \\ H_B & \text{, wenn } (P(H_F|M,c) = 0) \wedge (P(H_B|M,c) = 1) \\ H_{F\cup B} & \text{, wenn } (P(H_F|M,c) = 0) \wedge (P(H_B|M,c) = 0) \\ H_{F\cap B} & \text{, sonst} \end{cases} \tag{5.41}$$

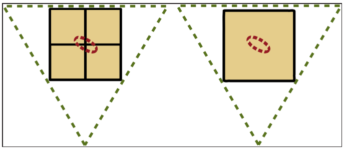
Für den Abfragealgorithmus, der die Menge der Zellen C zu einem Zustand zusammenfassen soll, ergibt sich folgende Funktion $\beta(C,M)$:

$$\beta(C,M) = \begin{cases} H_F & \text{, wenn } (\forall c \in C|\xi(c,M) = H_F) \\ H_B & \text{, wenn } (\exists c \in C|\xi(c,M) = H_B) \vee \\ & [(\exists c \in C|\xi(c,M) = H_{F\cup B}) \wedge (\exists c \in C|\xi(c,M) = H_{F\cap B})] \\ H_{F\cup B} & \text{, wenn } (\exists c \in C|\xi(c,M) = H_{B\cup F}) \wedge (\nexists c \in C|\xi(c,M) = H_B) \\ H_{F\cap B} & \text{, sonst} \end{cases} \tag{5.42}$$

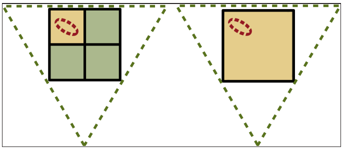
Der vorgestellte Algorithmus löst das ICDP formal exakt. Er basiert darauf, die Messungen als geometrische Formen in der Ebene zu betrachten und diese zu verknüpfen. Hierbei werden die jeweiligen Belegt- und Freimessungen zu jeweils einer Menge vereint. Aus dem Schnitt der beiden Vereinigungsmengen ergibt sich der Konfliktbereich und aus den Differenzmengen der Frei- bzw. Belegtbereich. Der Bereich, der durch keine der erzeugten Mengen abgedeckt wird, ist der unbekannte Bereich. Dieser Zusammenhang ist in Abbildung 5.19 verdeutlicht.

Um zu entscheiden, welchen Zustand eine Zelle hat, muss der Algorithmus die Gesamtmenge der Frei- und Belegtbereiche betrachten, welche durch die inversen Sensormodelle und Messungen erzeugt worden sind. Im Gegensatz zu bekannten Ansätzen existiert kein Zellzustand, der effizient rekursiv durch neue Messung aktualisiert wird. Eine Begrenzung der betrachteten Messungen ist somit notwendig, um eine obere Grenze bzgl. Laufzeitverhalten einhalten zu können.

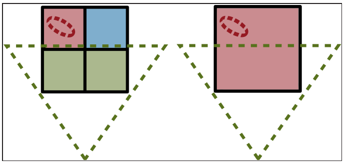
Auch sind je nach geometrischer Form der Frei- und Belegtbereiche mehr oder weniger komplexe Algorithmen notwendig, um die Vereinigungs-, Schnitt- oder Differenzmengen exakt zu bestimmen. Auf Basis solcher Algorithmen könnte man direkt den Belegungszustand eines Bereiches ermitteln, ohne den Umweg über die Zellen zu nutzen. Bei diesem Ansatz kann das Belegungsgitter als solches entfallen und es entsteht keine Diskretisierungsungenauigkeit. Die eigentliche Fusionierung findet bei



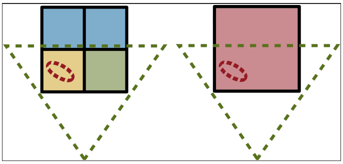
(a) Im einfachsten Fall haben alle Zellen den gleichen Zustand, welcher bei der Fusionierung erhalten bleibt



(b) Frei- und Konfliktbereiche werden zu einem Konfliktbereich fusioniert



(c) Ist eine Zelle belegt, so gilt dieses für den gesamten Bereich



(d) Konfliktbereiche und unbekannte Bereiche werden zu einem Belegtbereich aufgelöst

Abbildung 5.19.: Diskreter ICDP-Algorithmus: Freie, belegte, unbekannte und konfliktbehaftete Zellen (grün/rot/blau/gelb) werden durch Freibereiche (grünes gestricheltes Dreieck) und Belegtbereiche (rote gestrichelte Ellipse) erzeugt. Der ICDP-konforme Abfragealgorithmus $\beta(C)$ liefert für jedes Zellmengenpaar mit vier und einer Zelle jeweils das gleiche Ergebnis.

dieser Methode erst beim Abfragealgorithmus statt, während der Fusionsalgorithmus nur einen Messwertepuffer verwalten muss.

Beim vorgestellten Belegungsgitteransatz ist es umgekehrt: Die Fusionierung ist der weitaus komplexere Teil, während der Abfragealgorithmus nur wenige Fallunterscheidungen beim Betrachten der Zellzustände machen muss. Dadurch ist eine schnelle Bestimmung des Belegtzustandes vieler unterschiedlicher Zellmengen bei konstanter Messmenge möglich. Dieser Fall tritt beispielsweise bei der Pfadplanung eines Roboters ein, wo zu einem Zeitpunkt viele alternative Pfade geprüft werden müssen, ohne dass sich das Belegungsgitter zwischenzeitlich ändert. Das Belegungsgitter kann in diesem Fall als eine Art Zwischenspeicher betrachtet werden,

welches die Laufzeit des Algorithmus auf Kosten von Speicher und Genauigkeit bei wiederholter Anwendung senkt.

5.7.2. Probabilistischer Fusionsalgorithmus

In diesem Abschnitt wird ein Algorithmus vorgestellt, der im Gegensatz zum vorangegangenen Algorithmus probabilistisch arbeitet. Um eine Echtzeitfähigkeit für diesen zu ermöglichen, wird keine exakte Lösung des ICDP angestrebt, jedoch eine approximative. Auch werden Einschränkungen bzgl. der Genauigkeit bei den inversen Sensormodellen gemacht, um die Laufzeit des Algorithmus zu verringern.

Der Algorithmus unterstützt Belegtbereiche in Form von Punkten und Boxen und Freibereiche in Polygonform. Es werden unterschiedliche Messevidenzen und zusätzlich bei den Belegtbereichen Positionsunsicherheiten unterstützt. Zusammen mit dem später vorgestellten Abfragealgorithmus $\beta(X)$ formen diese Algorithmen eine approximative Lösung für das ICDP.

Zunächst werden die inversen Sensormodelle für die unterschiedlichen Messtypen vorgestellt. Diese generieren Evidenzen für einzelne Zellen.

Inverses Sensormodell für Freibereiche

Das inverse Sensormodell für Freibereiche generiert eine Evidenzverteilung aus einer Freimessung, welche auf unterschiedlichen Sensoren basieren kann (vgl. Kapitel 2). Freibereiche können sehr groß sein und somit viele Zellen enthalten. Sie können gleichzeitig komplexe Formen annehmen, insbesondere wenn Objekte den Freibereich unregelmäßig unterbrechen.

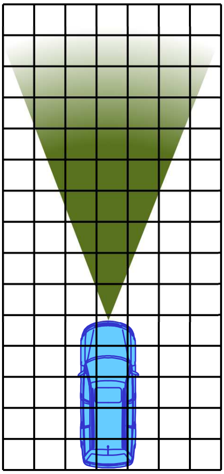
Ein geeignetes Modell scheint das Strahlenmodell zu sein, welches ausgehend vom Sensor einzelne Strahlen solange durch den Raum wirft, bis ein Belegtbereich diesen aufhält oder die maximale Sensorreichweite erreicht worden ist. Falls kein Belegtbereich erkannt worden ist, macht es insbesondere bei indirekt gemessenen Freibereichen Sinn, die Evidenz für die Freihypothese mit steigender Entfernung zu senken. Dies ist mit der unsicheren Schätzung der maximalen Sensorreichweite begründet, die von diversen unbekannten Faktoren abhängig ist, wie beispielsweise Reflektivität eines möglichen Ziels oder Witterungseinflüsse.

Im folgenden Modell wird ein Kernbereich von 0 bis R_{Min} definiert, in dem eine maximale Evidenz für einen Freibereich erreicht wird. Mit steigender Reichweite fällt die Freievidenz linear ab bis R_{Max} erreicht worden ist. Ab diesem Punkt ist die maximale Sensorreichweite erreicht worden, sodass

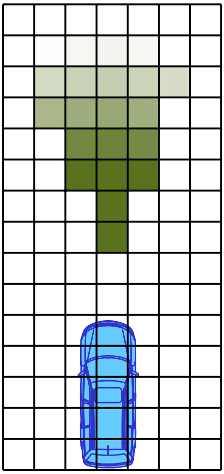
für weiter entfernte Zellen keine Freievidenz generiert wird. Dieses inverse Sensormodell findet soweit auch in [Weiss 2008] Verwendung.

Die Wahrscheinlichkeit $\rho_{H_F}(x,y)$, dass die Fläche der Zelle c mit den zum Sensor relativen Koordinaten x, y unter Berücksichtigung eines Sensor- bzw. Messzuverlässigkeitsfaktors τ frei ist, beträgt:

$$\rho_{H_F}(x,y) = \tau \cdot \max\left(0, \frac{R_{Max} - \max\left(\sqrt{x^2+y^2}, R_{Min}\right)}{R_{Max} - R_{Min}}\right) \quad (5.43)$$



(a) Freibereichssektor mit Kernbereich und radialen linearen Evidenzabfall wird über das Belegungsgitter gelegt



(b) Jede Zelle erhält die minimale Freievidenz der vollständig überlappenden Fläche

Abbildung 5.20.: Inverses Sensormodell für Freibereiche

In diesem Algorithmus werden nur solche Zellen aktualisiert, die vollständig im Sichtbereich des Sensors liegen und somit als komplett frei bestätigt werden können. Für die Entfernungsberechnung $\sqrt{x^2+y^2}$ sind die Koordinaten x und y innerhalb der Zelle so zu wählen, dass die Entfernung

maximiert wird. Diese Koordinaten befinden sich vom Sensor aus gesehen auf dem am weitesten entfernten Eckpunkt der Zelle. In Abbildung 5.20 ist die exemplarische Generierung von Freievidenzen für einzelne Zellen durch dieses inverse Sensormodell dargestellt.

Im Falle einer Reflexion eines Laserpunktes bzw. bei Erkennung eines Objektes durch einen Stereokamerasensor ist R_{Max} und R_{Min} so zu wählen, dass für die Entfernungsschätzung d der Messung gilt:

$$d = R_{Min} + \frac{1}{2}\left(R_{Max} - R_{Min}\right) \tag{5.44}$$

$R_{Max} - R_{Min}$ repräsentiert hierbei die Positionsunsicherheit des Objektes und somit indirekt die Unsicherheit über die Länge des Freibereiches. Durch die Linearität des Modells ist die Wahrscheinlichkeitsdichtefunktion bzgl. des Abstandes des Objektes zum Sensor dreiecksverteilt.

In diesem Fall ist bei indirekt messenden Sensoren eine Anhebung des Gewichtungsfaktors τ zusätzlich sinnvoll, da aufgrund der Objektdetektion die Wahrscheinlichkeit verringert worden ist, dass sich im Bereich zwischen Objekt und Sensor ein anderes Objekt befindet. Dieses ist durch die bekannte minimale Sensorreichweite und durch eine nicht erfolgte Verdeckung des detektierten Objektes begründet.

Das vorgestellte Freibereichsmodell ermöglicht die effiziente Berechnung von Evidenzen für bestimmte Freibereiche. Es unterstützt Strahlen-, Kegel- bzw. Sektormodelle, welche beispielsweise bei Laserscannern, Stereokameras und Ultraschallsensoren auftreten. Es ist somit durch entsprechende Parametrisierung modular für viele Sensoren einsetzbar.

Insgesamt basiert es darauf, eine Fläche mit variabler Evidenz zu generieren und diese den Zellen zuzuweisen. Um später das ICDP zu lösen, ist die Art unwichtig, wie diese Freifläche generiert wird. Entscheidend ist, dass die Freievidenz einer Zelle der minimalen Evidenz des überlappenden Bereiches entspricht. Es bleibt somit viel Raum, um alternative inverse Sensormodelle für Freibereiche zu generieren.

Inverses Sensormodell für punktförmige Objektmessungen

Für Messungen, die nur eine Positionsinformation aber keine Konturinformation wie Breite oder Länge liefern, ist ein Modell notwendig, welches punktförmige Belegungsinformationen in das Belegungsgitter fusioniert. Dieser Typ von Messungen wird beispielsweise häufig von Radarsensoren erzeugt.

Im Folgenden sei ein Objekt angenommen, welches neben Positionsinformationen p_x und p_y eine Schätzung der zugehörigen Varianzen σ_x und

σ_y aufweist. Die Varianzschätzung kann statisch für jeden Sensor festgelegt werden oder dynamisch aus einem Trackingalgorithmus bestimmt werden.

Neben den Varianzen muss wie beim Freibereichsmodell auch wieder ein Gewichtungsfaktor τ angegeben werden, der eine Gewichtung der Messung ermöglicht. Diese Gewichtung ist insbesondere bei widersprüchlichen Informationen wichtig: Damit kann die Sensordatenfusion dann entscheiden, welche Information glaubwürdiger ist.

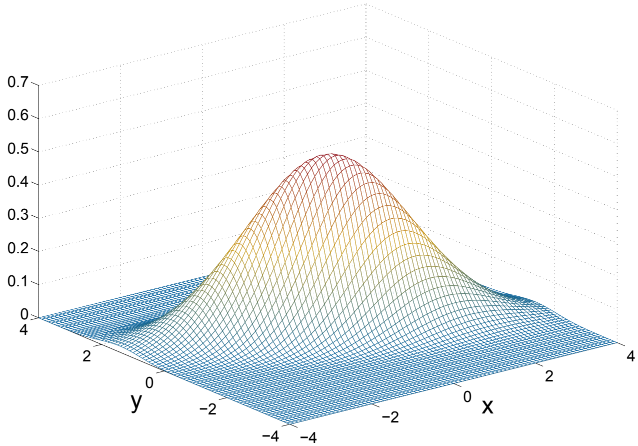


Abbildung 5.21.: Zweidimensionale Normalverteilung mit 45° Drehwinkel und unterschiedlichen Standardabweichungen in den einzelnen Dimensionen

Das Punktmodell $\rho_{P,H_B}(x,y)$ approximiert das Integral einer zweidimensionalen Normalverteilung: Die resultierende Belegtmasse einer Zelle entspricht dabei dem Volumen der Normalverteilung unter dieser Zelle. Abbildung 5.21 zeigt eine solche Normalverteilung mit einer möglichen Belegungsgitterstruktur.

Die Belegungswahrscheinlichkeit einer Zelle wird als Produkt der integralen Schnittflächen in x- und y-Richtung und Gewichtungsfaktor τ approximiert. Nach diesem Modell kann die Belegungswahrscheinlichkeit ρ_{P,H_B} einer Zelle mit den Koordinaten x, y wie folgt berechnet werden:

$$\rho_{P,H_B}(x,y) = \tau\, \rho'_{P,H_B}(x, p_x, \sigma_x)\, \rho'_{P,H_B}(y, p_y, \sigma_y) \tag{5.45}$$

Die Belegungswahrscheinlichkeit pro Dimension für eine Zelle der Größe $s \times s$ wird wie folgt berechnet:

$$\rho'_{P,H_B}(a,p,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{a-\frac{s}{2}}^{a+\frac{s}{2}} e^{-\frac{1}{2}\left(\frac{v-p}{\sigma}\right)^2} dv \tag{5.46}$$

In dem Punktmodell stecken zwei Vereinfachungen, sodass das Volumen nicht exakt, aber doch recht genau approximiert wird. Diese sind notwendig, um eine effiziente Berechnung der Masse unter Echtzeitbedingungen zu ermöglichen. Zum einen wird die Zelle, über die das Integral berechnet wird, um den Zellmittelpunkt so gedreht, dass diese senkrecht bzw. parallel zu den Achsen der Normalverteilung steht. Zum anderen wird nicht die zweidimensionale Normalverteilung integriert, sondern die eindimensionalen Normalverteilungen pro Dimension, die durch den Zellmittelpunkt verlaufen.

Die Massebestimmung erfordert insgesamt die Berechnung von zwei bestimmten Integralen der Normalverteilung. Da solche Integrale nicht geschlossen integrierbar sind, werden diese mittels nummerisch approximierten Tabellen angenähert. Durch geschickte Skalierung und Verschiebung lässt sich das Problem auf die bestimmte Integration einer Standardnormalverteilung bzw. auf die zweimalige Berechnung eines normierten Gaußschen Fehlerintegrals zurückführen. Letzteres wird in einer Tabelle in hoher Auflösung vorgehalten.

Inverses Sensormodell für rechteckförmige Objektmessungen

Neben punktförmigen Belegtmessungen gibt es Sensoren, die bereits bei der Signalvorverarbeitung Messpunktwolken so zusammenfassen, dass Objekte mit Konturen entstehen. Für weitere Informationen dazu sei auf Kapitel 2 verwiesen. Um für solche Sensoren ein inverses Sensormodell zu schaffen, wird im Folgenden ein Rechteckmodell verwendet.

Dieses inverse Sensormodell soll neben der Objektpositition und den Konturlängen auch probabilistische Faktoren nutzen. Hierzu wird die Varianz der Objektposition und wieder ein messungs- bzw. sensorabhängiger Gewichtungsfaktor τ berücksichtigt.

Das Rechteckmodell wird auf zwei eindimensionale Linienmodelle $\rho'_{R,H_B}(a,p,\sigma,l)$ entlang der Richtungsachsen der rechteckförmigen Messung reduziert. Die Linienmodelle betrachten die jeweilige Wahrscheinlichkeit, dass eine Zelle mit der Position a nicht mit einer Linie mit der Position (p_x, p_y) und der Länge (l_x, l_y) im eindimensionalen Fall überlappt. Die

Linien werden aus senkrechten Schnitten des Rechteckes generiert und durch je zwei Punkte $p-\frac{l}{2}$ und $p+\frac{l}{2}$ beschrieben. Die Zelle ist nicht belegt, wenn sich der linke Startpunkt der Linie rechts von der Zelle befindet oder wenn der rechte Endpunkt der Linie sich links von der Zelle befindet. Die Belegungswahrscheinlichkeit wird als dazu komplementäres Ereignis angesehen und folgendermaßen berechnet:

$$\rho'_{R,H_B}(a,p,\sigma,l) = \left(1-\frac{1}{\sigma\sqrt{2\pi}}\int\limits_{a+\frac{s}{2}}^{\infty} e^{-\frac{1}{2}\left(\frac{v-p+\frac{l}{2}}{\sigma}\right)^2} dv\right)\cdot \left(1-\frac{1}{\sigma\sqrt{2\pi}}\int\limits_{-\infty}^{a-\frac{s}{2}} e^{-\frac{1}{2}\left(\frac{v-p-\frac{l}{2}}{\sigma}\right)^2} dv\right) \tag{5.47}$$

Durch Umformung der Integrationsgrenzen lässt sich Gleichung 5.47 wie folgt vereinfachen:

$$\rho'_{R,H_B}(a,p,\sigma,l) = \frac{1}{2\pi\sigma^2}\int\limits_{-\infty}^{a+\frac{s}{2}} e^{-\frac{1}{2}\left(\frac{v-p+\frac{l}{2}}{\sigma}\right)^2} dv \int\limits_{a-\frac{s}{2}}^{\infty} e^{-\frac{1}{2}\left(\frac{v-p-\frac{l}{2}}{\sigma}\right)^2} dv \tag{5.48}$$

Das Rechteckmodell nimmt die Überlappungswahrscheinlichkeiten in X- und Y-Richtung als unabhängig an und modifiziert die beiden Linienmodelle mit dem Gewichtungsfaktor τ:

$$\rho_{R,H_B}(x,y) = \tau\, \rho'_{R,H_B}(x,p_x,\sigma_x,l_x)\, \rho'_{R,H_B}(y,p_y,\sigma_y,l_y) \tag{5.49}$$

Die Belegtevidenzverteilung eines beispielhaften Rechtecks für ein Belegungsgitter ist in Abbildung 5.22 dargestellt.

Das vorgestellte inverse Sensormodell lässt sich effizient berechnen: Wie beim punktförmigen inversen Sensormodell erfordert es die Produktbildung von vier Gaußschen Fehlerintegralen zusammen mit weiteren bekannten Faktoren. Ebenfalls wie beim Punktmodell werden diese jeweils auf das normierte Gaußsche Fehlerintegral zurückgeführt, welches in einer hoch aufgelösten Tabelle vorgehalten wird.

Das inverse Sensormodell berücksichtigt zwar einige probabilistische Faktoren, jedoch nicht die Varianz der Konturlängenschätzung. Diese kann man jedoch approximativ auf die Positionsunsicherheit aufschlagen. Wie beim inversen Sensormodell für punktförmige Belegtmessungen werden

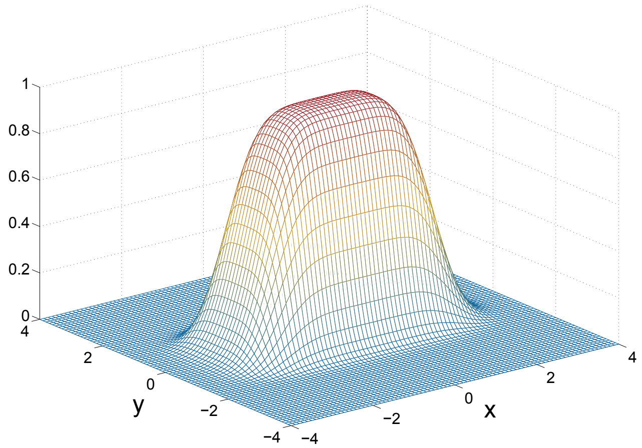


Abbildung 5.22.: Inverses Sensormodell für ein 4 × 2 großes Rechteck mit einer Positionsunsicherheit von $\sigma_x = 0,5$ bzw. $\sigma_y = 0,3$. Der Gewichtungsfaktor τ beträgt $1,0$.

die Zellen um den Zellmittelpunkt so gedreht, dass diese senkrecht zu den Achsen des Rechtecks liegen. Hierdurch entsteht ein weiterer Approximationsfehler.

Insgesamt wird jedoch eine gute Schätzung der Belegtwahrscheinlichkeit unter den Modellannahmen für eine einzelne Zelle erreicht. Problematischer wird später die genaue Schätzung vieler Zellen, die von einer rechteckförmigen Belegtmessung betroffen sind. Hier sind insbesondere bei hohen Varianzen und geringer Konturgröße stärkere Approximationsfehler zu erwarten.

Einbringung der Messevidenzen in eine Zelle

Um das ICDP zu lösen, ist die Modellierung der Abhängigkeiten und Unabhängigkeiten notwendig. Beim nichtprobabilistischen Algorithmus wurde dazu die betrachtete Menge der Messungen global zwischengespeichert, sodass kein Informationsverlust auftritt. Dieser Ansatz ermöglicht zwar eine exakte Rekonstruktion, ist aber sehr aufwendig und je nach Größe der betrachteten Messmenge auch nicht echtzeitfähig.

Eine andere Möglichkeit besteht darin, für die jeweilige Zelle an Position x,y ihre zugeordnete Evidenzmasse dort lokal zu speichern. Hierzu

muss zunächst für eine Messung m_k das passende inverse Sensormodell bestimmt werden:

$$\rho(m_k,c) = \begin{cases} \rho_{H_F}(x,y) & \text{, wenn } m_k \text{ eine Freimessung ist} \\ \rho_{P,H_B}(x,y) & \text{, wenn } m_k \text{ eine punktförmige Belegtmessung ist} \\ \rho_{R,H_B}(x,y) & \text{, sonst} \end{cases} \tag{5.50}$$

Neben der Evidenzmasse wird die Information benötigt, die eine Zuordnung der Messevidenz zu einer Messung erlaubt. Hierzu wird eine einzigartige Kennung pro Messung $id(m_k)$ erzeugt. Diese Kennung wird aus einer Sensoridentifikationsnummer und einem fortlaufenden Zähler generiert. Es ist somit möglich, aus solch einer Messkennung sowohl den Typ des inversen Sensormodells als auch den Sensor wieder zu bestimmen.

Der Fusionsalgorithmus $\xi(c,M^k)$ generiert für die Zelle c aus k Messungen $M^k = \{m_1, \dots m_k\}$ Tupel, die aus der Zellkoordinate (x, y), einer Messkennung id und einer Evidenz $\rho(m_i,c)$ bestehen und fügt sie dem Ringpufferspeicher dieser Zelle hinzu[6]. Dieser Speicher besitzt eine maximale Kapazität von n Einträgen, sodass bei Überschreitung die ältesten Einträge aus diesem herausfallen:

$$\xi(c,M^k) = \bigcup_{i=\max(1,k-n+1)}^{k} (x,y,id(m_i),\rho(m_i,c)) \tag{5.51}$$

Der Algorithmus kann entgegen der mathematischen Formulierung effizient umgesetzt werden, sodass die Aktualisierung des Zellzustandes mit einer Messung in konstanter Zeit möglich ist. Ein beispielhafter Inhalt zweier Zellen ist in Tabelle 5.1 dargestellt.

Da alte Messungen automatisch überschrieben werden, wenn der Ringpuffer voll ist, ist indirekt ein Alterungsmechanismus in diesem Algorithmus integriert. Dieser ist hilfreich, um bei der Dezert-Smarandache Theorie die Konfliktmasse des Zellzustandes potenziell zu reduzieren. Hierzu müssen solange neue widerspruchsfreie Messungen die Zelle treffen, bis die widersprüchlichen Messungen aus dem Ringpufferspeicher herausgefallen sind.

Insgesamt ermöglicht dieser Fusionsalgorithmus eine weitgehende Rekonstruktion der Abhängigkeiten und Unabhängigkeiten. So lässt sich für

[6] Die Zellkoordinaten in den Tupeln werden später für eine einfachere mathematische Darstellung benötigt. Die Softwareimplementation kann diese aus Zellindeces ermitteln, sodass diese die Zellkoordinaten in den Tupeln nicht benötigt.

Sensor-ID	Mess-ID	Evidenz
1 (P,H_B)	01	0,1
2 (H_F)	01	0,1
3 (R,H_B)	01	0,3
2 (H_F)	02	0,1

Sensor-ID	Mess-ID	Evidenz
1 (P,H_B)	01	0,15
2 (H_F)	01	0,1
3 (R,H_B)	01	0,4
2 (H_F)	03	0,2

Tabelle 5.1.: Beispielhafter Inhalt der Ringpufferspeicher zweier Zellen (ohne Zellkoordinaten)

jede Messung, deren Evidenzen noch aus keinem Ringpufferspeicher herausgefallen sind, die ursprüngliche Massenverteilung gemäß dem inversen Sensormodell approximativ wiederherstellen. Die nicht exakte Lösung ist durch die Diskretisierung der Zellen begründet und tritt auf, wenn eine Messung mittels inversen Sensormodells eine nicht homogene Evidenz bzgl. der Zellfläche liefert.

Nachteilig an dem Ansatz ist, dass die Zelle maximal nur soviel Evidenz sammeln kann, wie der Ringpufferspeicher an Messungen aufnehmen kann. Dieser Nachteil ist je nach Ringpuffergröße hinsichtlich der IMDP-Annahme jedoch zu vernachlässigen. Diese nimmt an, dass Messungen aus einem Sensor korrelieren und somit eine Abhängigkeit aufweisen. Daraus folgt, dass wenn eine Hypothese bereits von mehreren Messungen eines Sensors bestätigt worden ist, weitere ähnliche Messungen nur eine geringe zusätzliche Evidenz liefern. Wenn man die Messreihe dann rückwärts betrachtet, so liefern die ältesten Messungen im Durchschnitt die geringsten zusätzlichen Evidenzen und können am ehesten entfallen.

Abfragealgorithmus

Der Fusionsalgorithmus $\xi(c,M^k)$ im vorherigen Abschnitt wurde so entworfen, dass er eine approximativ ICDP-konforme Abfrage eines Bereiches ermöglicht. Der zugehörige Abfragealgorithmus wird in diesem Abschnitt vorgestellt. Hierzu ist eine mathematische Verbunddarstellung des Belegungsgitters notwendig:

Der Abfragealgorithmus $\beta(Z)$ soll im Folgenden eine Befahrbarkeitsschätzung für eine Zellmenge nach der Dezert-Smarandache Theorie bestimmen. Dazu sei angenommen, dass eine rechteckige Fläche von x_{min} und y_{min} bis x_{max} und y_{max} bestehend aus einer Menge von Zellen Z' berücksichtigt werden muss, um eine Befahrbarkeitswahrscheinlichkeit einer

gegebenen Trajektorie bestimmen zu können[7]. Wie bereits im vorherigen Abschnitt erwähnt, verwaltet jede Zelle mehrere Tupel in einem Ringpufferspeicher, welche jeweils aus einer Kennung $id \in I$, einer Evidenzmasse $\rho \in E$ und der Zellkoordinate $(x,y) \in X \times Y$ bestehen:

$$X = \{x_{min}, x_{min}+s, x_{min}+2s, \dots, x_{max}\} \tag{5.52}$$

$$Y = \{y_{min}, y_{min}+s, y_{min}+2s, \dots, y_{max}\} \tag{5.53}$$

$$I = \mathbb{N} \tag{5.54}$$

$$E = [0,1] \tag{5.55}$$

$$Z' = X \times Y \times I \times E \tag{5.56}$$

Die Größe der Ringpufferspeicher jeder Zelle ist auf n Einträge begrenzt, sodass der unendlich große Zustandsraum Z' auf einen Zustandsraum Z mit endlich vielen Elementen reduziert werden kann. Zusätzlich gilt, dass die Kennung id für jede Messung eindeutig ist und somit in jedem Ringpufferspeicher höchstens einmal auftritt. Für die Elemente in $Z \subset Z'$ kann somit folgende Relation R definiert werden:

$$\begin{aligned}&(Z \in R) \to \forall (x,y,id,\rho) \in Z \ : \\ &\quad (|Z \cap (x \times y \times I \times E)| \leq n) \wedge (|Z \cap (x \times y \times id \times E)| \leq 1)\end{aligned} \tag{5.57}$$

Die Ringpufferspeicher der Zellen in diesem Bereich werden durch verschiedene Messungen und Messtypen gemäß dem vorherigen Abschnitt 5.7.2 aktualisiert. Abhängig vom Messtyp müssen verschiedene Abfragemechanismen gewählt werden. Der Messtyp lässt sich aus der Dekodierung der ID bestimmen.

Wenn das Punktmodell verwendet worden ist, kann eine fusionierte Belegungsschätzung pro Messung durch einfaches Summieren aller Teilevidenzen der entsprechenden Zellen berechnet werden. Bei der Verwendung des Boxmodells approximiert sich die Belegungsevidenz dieser Fläche pro Messung durch Maximum-Bildung, welches eine untere Grenze der Belegungssschätzung darstellt (vgl. Gleichung 5.49):

[7] Zur einfacheren mathematischen Darstellung wird angenommen, dass die Fläche rechteckig ist und sich senkrecht zum Belegungsgitter befindet. Für die Umsetzung des Abfragealgorithmus $\beta(Z)$ ist es jedoch irrelevant, welche Form die Zellmenge Z impliziert: So wird in der Softwareimplementation dafür eine allgemeine Polygonbeschreibung benutzt.

$$\Phi'_{H_B}(Z,id) = \begin{cases} \sum\limits_{\forall(x,y,id,\rho)\in(X\times Y\times id\times E)\cap Z} \rho & \text{, wenn } id \text{ Modell } \rho_{P,H_B} \text{ impliziert} \\ \max\limits_{\forall(x,y,id,\rho)\in(X\times Y\times id\times E)\cap Z} \rho & \text{, wenn } id \text{ Modell } \rho_{R,H_B} \text{ impliziert} \\ 0 & \text{, sonst} \end{cases} \tag{5.58}$$

Als Ergebnis erhält man eine Belegungsevidenz der Messung mit der Kennung id für den Bereich der Zellmenge Z: Hierbei handelt es sich approximativ um die Gewichtung τ der Messung multipliziert mit der Wahrscheinlichkeit, dass diese Messung unter Berücksichtigung der Positionsunsicherheit den abgefragten Bereich belegt.

Das Zusammenfassen der Belegtmassen erfolgt nach der Dezert-Smarandache Theorie, wobei eine optimierte Variante möglich ist, da keine Konflikte mit Freimassen auftreten. Es entstehen somit nur die beiden Evidenzen für die Belegt- und Unbekannthypothese. Die Optimierung basiert darauf, ausschließlich die Abnahme der Unbekanntmasse zu ermitteln und die Belegtmasse komplementär zu betrachten. Letztere berechnet sich dann nach:

$$\Phi_{H_B}(Z) = 1 - \prod_{i=1}^{\infty} \left(1 - \Phi'_{H_B}(Z,i)\right) \tag{5.59}$$

Für das Freibereichsmodell müssen zunächst die Freievidenzen $\Phi'_{H_F}(c)$ nach der Dezert-Smarandache Theorie für einzelne Zellen $c \in Z$ im abgefragten Bereich bestimmt werden. Auch hier ist aufgrund der konfliktfreien Massen die optimierte Variante möglich:

$$\Phi'_{H_F}(c) = 1 - \prod_{\forall(x,y,id,\rho)\in c} \begin{cases} (1-\rho) & \text{, wenn } id \text{ Modell } \rho_{H_F} \text{ impliziert} \\ 1 & \text{, sonst} \end{cases} \tag{5.60}$$

Die Zelle mit der geringsten Freievidenz der Zellmenge repräsentiert die Freievidenz der gesamten Fläche:

$$\Phi_{H_F}(Z) = \min_{\forall c\in Z} \left(\Phi'_{H_F}(c)\right) \tag{5.61}$$

Insgesamt erhält man aus allen Messungen eine kombinierte Freievidenz für den spezifischen Bereich, wobei die Unsicherheiten des inversen Sensormodells für Freibereiche berücksichtigt worden sind. Hierbei wurde die geometrische Form der Messungen approximativ auf Zellgrößen diskretisiert und dessen Evidenzen vereint, sodass je nach Auflösung des

Belegungsgitters mehr oder weniger Approximationsungenauigkeiten zu erwarten sind.

Um eine endgültige Massenverteilung für den abzufragenden Bereich zu generieren, werden die beiden erzeugten Evidenzen $\Phi_{H_F}(Z)$ und $\Phi_{H_B}(Z)$ in Massenverteilungen konvertiert und als Eingabewert für die Dezert-Smarandache Theorie genutzt:

$$\beta(Z) = DSmT\left(\langle\Phi_{H_F}(Z), 0, 1-\Phi_{H_F}(Z), 0\rangle, \langle 0, \Phi_{H_B}(Z), 1-\Phi_{H_B}(Z), 0\rangle\right) \quad (5.62)$$

Da die Fusion von widersprüchlichen Massenverteilungen mittels Dezert-Smarandache Theorie nicht kommutativ ist, wurden zuerst alle Messungen für Belegt- und Freibereiche zu einer konfliktfreien Massenverteilung zusammengefasst. Somit ist der vorgestellte Fusionsalgorithmus approximativ kommutativ. Die Approximationseinschränkung ergibt sich aus dem endlich großen Ringpufferspeicher der einzelnen Zellen.

Sensor-ID	Mess-ID	Evidenz A
1 (P, H_B)	01	0,25
3 (R, H_B)	01	0,4
* (H_F)	*	0,19

Tabelle 5.2.: Zwischenergebnis des Abfragealgorithmus β nach Traversierung der Ringpufferspeicher der beiden Zellen aus Tabelle 5.1.

Insgesamt fasst der Abfragealgorithmus $\beta(Z)$ abhängige Zellinformationen zusammen, die durch einzelne Messungen verursacht wurden. Die einzelnen Messungen hingegen werden zunächst als unabhängig betrachtet. Ein Zwischenergebnis einer beispielhaften Menge ist in Tabelle 5.2 dargestellt.

Ist der abzufragende Bereich so klein gewählt, dass er nur eine Zelle umfasst, so lässt sich wie bei klassischen Belegungsgittern der Zustand einzelner Zellen abfragen. In Abbildung 5.23 sind diese Einzelevidenzen als transparentes Bitmap über dem Umgebungsbild dargestellt. Zusätzlich wird in diesem Beispiel die Befahrbarkeitswahrscheinlichkeit des Fahrschlauches in bis zu 16 m Entfernung bestimmt und visualisiert. Hierbei wurden Hunderte Zellen gemäß dem probabilistischen Algorithmus zu einer Gesamtwahrscheinlichkeit fusioniert.

Der Abfragealgorithmus liefert eine Massenverteilung, welche verschiedene Szenarien repräsentieren kann. Der häufigste und einfachste Fall



Abbildung 5.23.: Belegungsgitter der einzelnen Zellzustände in einer innerstädtischen Verkehrsszene. Die Befahrbarkeitswahrscheinlichkeit des Fahrschlauchs wird durch die beiden magentafarbenen Dreiecke dargestellt.

liegt in einer hohen Evidenzmasse für Belegt- bzw. Freibereiche, welche einen entsprechend belegt bzw. freien Raum darstellen.

Falls die Masse für den Unbekanntzustand hoch ist, weist dies auf mindestens eine Zelle hin, für die keine oder nur eine geringe Messevidenz vorliegt. Gleichzeitig sind keine oder nur geringe Belegtevidenzen für andere Zellen aufgetreten.

Ein hoher Konfliktwert deutet auf widersprüchliche Sensorinformationen hin: So wurde mindestens eine Zelle im abgefragten Bereich von einer oder verschiedenen Messungen als frei und gleichzeitig als belegt gemessen und es existiert keine weitere Zelle mit einer hohen Belegt- oder Unbekanntevidenz. Daraus folgt, dass zu diesem Zeitpunkt keine zuverlässige Aussage über diesen Bereich gemacht werden kann. Muss jedoch eine diskrete Entscheidung gefällt werden, so kann man in diesem Fall das Verhältnis der Frei- und Belegtmassen betrachten.

5.7.3. Belegungsgitter mit variabler Auflösung

Der im vorherigen Abschnitt vorgeschlagene Algorithmus ist durch die Verwendung von Ringpufferspeichern in den einzelnen Zellen sehr speicheraufwendig. Eine Möglichkeit dem zu begegnen, ist die Zellauflösung des Belegungsgitters variabel zu gestalten. Diese Grundidee wird soweit auch in [Montemerlo 2004] und [Ferguson 2006] verfolgt.

Die Motivation dieser Arbeiten ist, dass in größerer Entfernung die Positionsunsicherheit von detektierten Hindernissen zunimmt und somit eine hohe Auflösung des Belegungsgitters in großer Entfernung überflüssig erscheint. Zudem wird es Pfadfindungsalgorithmen erleichtert, eine Trajektorie durch das Umfeld zu finden, da zum einen weniger Zellen zu überprüfen sind und zum anderen die Auftretenswahrscheinlichkeit von einzelnen vermeintlichen Löchern über lang gestreckte Hindernisse reduziert wird. Ein Beispiel für solch ein Belegungsgitter ist in Abbildung 5.24 dargestellt.

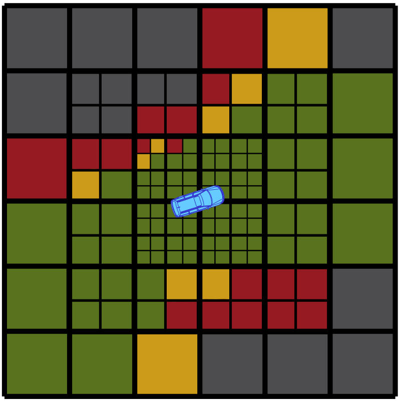


Abbildung 5.24.: Schematisches Beispiel für ein dreischichtiges Belegungsgitter mit unterschiedlichen Zellgrößen

In [Montemerlo 2004] werden zur Erzeugung eines solchen Gitters mehrere Belegungsgitter in Schichten übereinandergesetzt. Das Belegungsgitter mit der höchsten Auflösung hat die kleinste Größe und erstreckt sich über den Nahbereich, während das Belegungsgitter mit der niedrigsten Auflösung den maximalen Aktionsradius des Fahrzeugs repräsentiert. In [Montemerlo 2004] wird dieser Aufbau eine „Pyramide von Belegungsgittern“ genannt.

Diese mehrschichtigen Belegungsgitter haben das Problem, dass, wenn

sich das Fahrzeug bewegt, es zu Zellinitialisierungen in den einzelnen Schichten kommt. Hierbei entstehen Bereiche mit unbekanntem Belegungszustand auf den Rändern der einzelnen Belegungsgitter (vgl. blaue Zellen in Abbildung 5.25). Für das Belegungsgitter mit der größten Auflösung ist dies akzeptabel, nicht jedoch für die kleineren. Um die neu hinzugekommenen Zellen geschickt zu initialisieren, könnten beispielsweise die Zellinformationen aus dem nächstgrößeren Belegungsgitter zurate gezogen werden. Hierzu müsste die Evidenz einer großen Zelle auf mehrere kleinere übertragen werden.

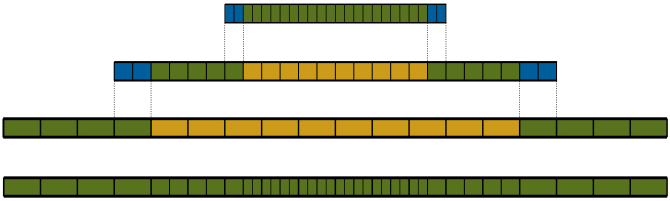


Abbildung 5.25.: Schnitt durch ein dreischichtiges Belegungsgitter. Die blauen Zellen werden für die Übergabe der Zellinitialisierungen (gestrichelte Linien) benötigt und stehen wie die zu niedrig aufgelösten gelben Zellen dem eigentlichen Belegungsgitter (unten) nicht zur Verfügung.

Bei der Aufteilung oder beim Zusammenfassen der Evidenzen einer Zellmenge sollten außer den Diskretisierungsungenauigkeiten möglichst keine weiteren Informationen verloren gehen. Hierzu ist eine formale Definition einer konsistenten Abbildungsoperation $\omega(C)$ hilfreich:

Definition 6 (Konsistente Verschiebung von Belegungsgittern)**.**
Seien $\xi(X,M)$ und $\beta(X)$ Algorithmen, die das ICDP gemäß Definition 4 lösen. Die Abbildung $\omega(X) \rightarrow Y$ einer Zellmenge X auf die Zellmenge Y ist im Rahmen einer Belegungsgitterverschiebung genau dann konsistent, wenn $\beta(\xi(\omega(\xi(C_1,M_1)) \cup \xi(C_2,M_1),M_2)) = \beta(\xi(\xi(C_1 \cup C_2,M_1),M_2))$ für alle möglichen disjunkten Zellmengen C_1, C_2 und alle Messungen M_1 und M_2 ist.

Da bei einer Verschiebungsoperation die Größe des Bereiches gleich bleibt, den die Zellen X bzw. Y repräsentieren, erfüllt ω diese Definition, wenn es die Fähigkeit von $\xi(X,M)$ und $\beta(X)$ zum Lösen des ICDP im Allgemeinen nicht einschränkt. Aus der Definition lassen sich folgende Eigenschaften für einen konsistenten Abbildungsoperator ableiten:

- $C_2 = \emptyset$ und $M_2 = \emptyset$: Die neue Zellmenge $\omega(C_1)$ muss das gleiche Abfrageergebnis liefern, wie die alte Zellmenge C_1.
- $C_2 = \emptyset$ und $M_2 \neq \emptyset$: Die neue Zellmenge $\omega(C_1)$ verhält sich bzgl. Abfrageergebnis bei weiteren Messungen wie die alte Zellmenge C_1.
- $C_2 \neq \emptyset$ und $M_2 = \emptyset$: Wenn die neue Zellmenge $\omega(C_1)$ und die alte Zellmenge C_1 mit weiteren Zellen vereinigt werden, liefert der Abfragealgorithmus für beide Zellmengen das gleiche Ergebnis.
- $C_2 \neq \emptyset$ und $M_2 \neq \emptyset$: Die beiden Mengen liefern auch das gleiche Ergebnis, wenn sie durch neue Messungen aktualisiert werden.

Beim Bayesansatz wie ihn [Montemerlo 2004] verwendet, kann der Zellzustand identisch auf kleinere Zellen übertragen werden. Dieses Vorgehen liefert aber bei dem probabilistischen Algorithmus, welcher in den vorherigen Abschnitten vorgestellt worden ist, kein konsistentes Ergebnis.

Im Folgenden wird ein Abbildungsoperator für diesen Algorithmus vorgeschlagen, der eine Konsistenz nach Definition 6 aufweist. Hierzu muss die Evidenz einer großen Zelle d auf den Zustand von $n \times n$ Zellen $c_{i,j} \in C$ je nach Messtyp aufgespalten oder übertragen werden.

Die Abbildungsoperation basiert darauf, die Freievidenzen und die Evidenzen für rechteckförmige Messungen identisch zu übernehmen: Hier ist die Konsistenz durch den $\min$- bzw. $\max$-Operator begründet. Bei punktförmigen Belegtevidenzen, welche durch den Summationsoperator fusioniert worden sind, wird die Evidenz gleichmäßig auf alle Zellen aufgespaltet:

$$\omega'(d) = \bigcup_{\forall (x,y,id,\rho)\in d} \begin{cases} (id, \frac{\rho}{n^2}) & \text{, wenn } id \text{ Modell } \rho_{P,H_B} \text{ impliziert} \\ (id, \rho) & \text{, sonst} \end{cases} \tag{5.63}$$

Der Abbildungsoperator $\omega(d)$ erzeugt n^2 ähnliche Zellen. Diese unterscheiden sich lediglich durch die Zellkoordinaten (x,y), welche die Hilfsfunktion $\mathrm{pos}(c)$ für eine Zelle c als Tupel zurückliefert:

$$\omega(d) = \bigcup_{\forall c_{i,j}\in C} \left(\mathrm{pos}(c_{i,j}) \times \omega'(d)\right) \tag{5.64}$$

Der vorgeschlagene probabilistische Algorithmus unterstützt Belegungsgitter mit variabler Auflösung, indem die Evidenztabelle mit geringen Verlusten auf andere Zellen übertragen werden kann. Um die Informationsverluste zu minimieren, sollten die Zellgrößen aller Belegungsgitter so gewählt werden, dass sie einem ganzzahligen Vielfachen von dem Belegungsgitter

mit der nächstniedrigeren Auflösung entsprechen. In diesem Fall ist nur ein Diskretisierungsverlust zu verzeichnen, welcher bei Belegungsgittern mit variabler Auflösung prinzipbedingt aber in Kauf genommen werden muss.

Neben den theoretischen Grundlagen ist in diesem Abschnitt ein insgesamt recht einfacher ω-Algorithmus vorgestellt worden. Dieser erlaubt zwar die konsistente Verwendung von Belegungsgittern mit variabler Auflösung und probabilistischen Fusions- und Abfragealgorithmus, ist jedoch bzgl. Speicherbedarf und Laufzeit noch nicht optimal. So ist eine Erweiterung denkbar, die es erlaubt, viele zu niedrig aufgelöste Zellen weder aktualisieren noch vorhalten zu müssen. Die Grundidee ist, dass der Zustand solch einer Zelle im Prinzip schon redundant durch eine Menge höher aufgelöster Zellen vorliegt. Im Beispiel von Abbildung 5.25 entsprechen solche redundanten Zellen den gelben Zellen.

Um den Zellzustand der großen Zelle zu erzeugen, ist ein Algorithmus notwendig, der die Evidenz von vielen kleinen Zellen auf diese überträgt. Für den vorgestellten probabilistischen Algorithmus ist dieses nur approximativ möglich, da es denkbar ist, dass der Inhalt vieler Ringpufferspeicher auf einen Ringpufferspeicher mit jeweils der gleichen Größe nicht verlustfrei übertragen werden kann. Dieser potenzielle Informationsverlust ist jedoch auf die prinzipbedingte und gewollte höhere Diskretisierung zurückzuführen und ist damit akzeptabel.

Ziel des Algorithmus muss es somit sein, die Evidenzen der Zellmenge auf die große Zelle so zu übertragen, dass diese Zelle höchstens diesen Diskretisierungsverlust erleidet. Dies wird erreicht, wenn diese Zelle den gleichen Zustand annimmt, als wenn sie direkt vom probabilistischen Algorithmus mit der gesamten Messmenge aktualisiert worden wäre.

Ein solcher Algorithmus ist approximativ möglich, wenn sich aus den Kennungen der Ringpufferspeichereinträge eine zeitliche Abfolge der Messungen ableiten lässt. Auf eine genaue Darstellung dieses Algorithmus sei hier verzichtet, da sich dieser dem bereits vorgestellten Abfragealgorithmus $\beta(X)$ in großen Teilen ähnelt.

5.7.4. Zusammenfassung

Wie in vorherigen Abschnitten vorgestellt, gibt es in der Literatur keine bekannten Belegungsgitteransätze, die die Abhängigkeiten zwischen Zellzuständen gemäß der ICDP-Definition systematisch auflösen. Deshalb wurden in diesem Abschnitt zwei Algorithmen vorgestellt, die dieses Problem berücksichtigen.

Es konnte zwar kein probabilistischer Fusions- und Abfragealgorithmus

geschaffen werden, der dieses Problem in Echtzeit formal löst. Es ist jedoch eine approximative Lösung geschaffen worden, welche deutlich bessere Eigenschaften zeigt als die bekannten Ansätze. So ist diese Lösung beispielsweise in der Lage, die Abhängigkeiten im Beispiel von Abbildung 5.17 exakt aufzulösen.

Der vorgestellte probabilistische Algorithmus enthält verschiedene inverse Sensormodelle, die zusammen mit dem Fusions- und Abfragealgorithmus je nach Modell, Messung und Zellauflösung einen mehr oder weniger großen Fehler bzgl. der ICDP-Definition verursachen.

So werden Punktziele sehr gut approximiert und der Freibereich je nach Stärke von Evidenzunterschieden innerhalb einer Zelle und Zellgröße auch gut angenähert. Rechteckförmige Messungen weisen je nach Konturgröße und Varianz jedoch einen mittleren bis großen Fehler auf. Letzteres ist damit begründet, dass zu wenig Metainformationen in ein einzelnes Evidenzattribut passen, um eine genaue Abhängigkeitsanalyse solch einer Zellmenge zu generieren. Hier wurde nur die Belegungswahrscheinlichkeit der einzelnen Zelle berechnet, ohne zu berücksichtigen, dass aufgrund der statischen Größe der Objektkontur sich weitere Abhängigkeiten ergeben. Dennoch ist aufgrund typischer Fahrschläuche der Fehler in der Praxis deutlich geringer als theoretisch möglich. So maximiert sich der Fehler, wenn die abzufragende Zellmenge dieses rechteckige Objekt in geringem Abstand möglichst vollständig umfließt, das Objekt möglichst klein und gleichzeitig eine hohe Positionsvarianz aufweist. Dieses tritt jedoch in Anwendungen praktisch nie auf.

Ein weiterer Approximationsfehler tritt durch die Größenbegrenzung des Ringpufferspeichers auf: Sobald dessen Kapazität erreicht ist, fallen die ältesten Messungen und deren Evidenzen aus der Bewertung. Dies kann bei widersprüchlichen Informationen, wie sie beispielsweise dynamische Objekte verursachen, aber auch hilfreich sein: So wird eine ehemals belegte Zelle nach einigen Messungen vollständig wieder freigegeben, was beispielsweise in Abbildung 5.26 verdeutlicht wird. Auch liefert unter der IMDP-Annahme eine ältere Messung nur eine geringe zusätzliche Evidenz, wenn bereits viele neuere vorliegen.

Insgesamt ist der Approximationsfehler trotz diverser negativer Einflüsse sehr gering verglichen mit konventionellen Ansätzen. Fast noch wichtiger aus Applikationssicht ist jedoch, dass das Belegungsgitter jetzt eine sinnvolle Schnittstelle liefert, den Umgebungszustand abzufragen: Die Applikation kann mittels einer allgemeinen Polygonbeschreibung die Befahrbarkeitswahrscheinlichkeit eines beliebigen Korridors ermitteln, wobei die einzelnen Zellen und deren Zustände vollständig verborgen bleiben. Bei konventionellen Ansätzen wird eine Menge von Einzelzellzuständen

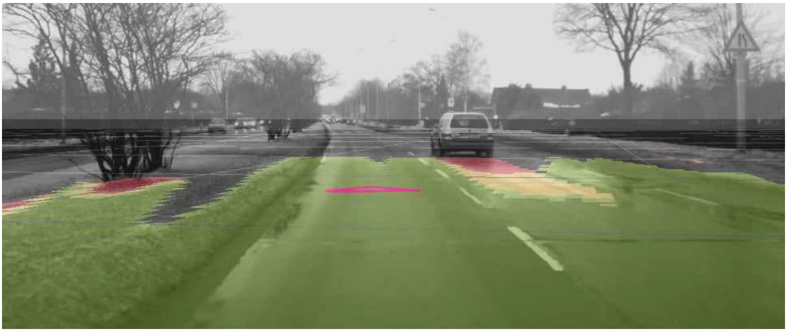


Abbildung 5.26.: Dynamische Objekte wie fahrende Autos verursachen Konfliktbereiche (gelb). Diese werden freigegeben, sobald die alten Belegtmessungen durch neue Freimessungen aus dem Ringpufferspeicher fallen.

zurückgeliefert, was eine eher künstliche und wenig anwendungsgerechte Beschreibung eines Korridors darstellt und zu stärkeren Ungenauigkeiten führt.

Da bei diesem Ansatz jede Zelle einen kleinen Ringpufferspeicher besitzt, ist der Speicherbedarf dieses Belegungsgitters signifikant größer verglichen zu bekannten Ansätzen. Die Fusionierung von Messungen in Zellzuständen ist sehr schnell und kann in konstanter Zeit erfolgen. Im Gegensatz dazu ist die Abfrage von Bereichen aufwendiger, da bei allen betroffenen Zellen die kompletten Ringpufferspeicher abgearbeitet werden müssen.

Um sowohl dem hohen Speicherbedarf als auch der hohen Rechenzeit bei der Ausführung des Abfragealgorithmus entgegenzuwirken, sind verschiedene Maßnahmen getroffen worden: Zum einen sind nur sehr einfache Operationen notwendig, um die Ringpufferspeicher zu verarbeiten. Diese bestehen im Wesentlichen aus dem Minimum-, Maximum- und Additionsoperator. Zum anderen wurde vorgeschlagen, ein Belegungsgitter mit variabler Auflösung zu verwenden. Im Vergleich zu einem großen hoch aufgelösten Belegungsgitter senkt dieser Ansatz neben dem Speicherbedarf auch die Laufzeit, da in einem abgefragten Fahrschlauch sich tendenziell weniger Zellen befinden.

Anstatt Belegungsgitter zu verwenden, ist es alternativ denkbar, dass man alle Messungen in einem großen Ringpufferspeicher ablegt. Der Abfragealgorithmus berechnet dann den Schnitt dieser Messungen mit ei-

nem Polygon, welches beispielsweise den Fahrschlauch repräsentiert und fasst diese mittels Logikalgorithmen zusammen. Das Ergebnis ist annähernd das Gleiche wie beim vorgeschlagenen probabilistischen Algorithmus. Da keine Diskretisierung und keine Vereinfachungen stattfinden, ist dieser Ansatz genauer und verbraucht auch signifikant weniger Speicher abhängig von den Ringpuffergrößen. Allerdings sind Algorithmen, die in der Lage sind, konkave Polygone mit mehrdimensionalen normalverteilten Wahrscheinlichkeitsdichtefunktionen, unscharfen Objektkonturen und Freibereichspolygonen mit variabler Evidenz exakt zu schneiden und auszuwerten, schwer zu entwickeln. Außerdem brauchen diese wohl eine deutlich höhere Laufzeit, was Zweifel an der Echtzeitfähigkeit solcher Algorithmen bei größeren Messmengen aufkommen lässt.

Von diesem Standpunkt aus gesehen ähnelt der vorgestellte Belegungsgitteransatz einer Art Zwischenspeicher, mit dem man eine geringere Laufzeit gegen erhöhten Speicherbedarf und geringere Genauigkeit tauscht.

5.8. IMDP-Algorithmen

In diesem Abschnitt werden die Abhängigkeiten, die zwischen verschiedenen Messungen auftreten können, behandelt. Ziel ist es, geeignete Methoden zur Auflösung dieser Abhängigkeiten auf den im vorherigen Abschnitt vorgestellten probabilistischen Belegungsgitteralgorithmus zu übertragen.

Aus der IMDP-Definition ergibt sich, dass mindestens die Genauigkeitsanforderung im Sinne der Applikation berücksichtigt werden muss, um entscheiden zu können, ob ein Algorithmus zusammen mit einer gegebenen Sensorik das IMDP löst. Somit existiert hierfür weder eine allgemeingültige Lösung, noch ist es zwingend mit einer gegebenen Sensorik und Genauigkeitsanforderung immer lösbar. Letzteres ist beispielsweise der Fall, wenn die Applikation praktisch Ground Truth-Daten von der Sensordatenfusion fordert, die Sensorik jedoch nicht annähernd ideale Messdaten liefert.

Insgesamt muss ein IMDP-Algorithmus in der Gesamtheit des kompletten Systems betrachtet werden, sodass dieser Abschnitt nur Vorschläge für solche Algorithmen zeigt, ohne diese genau zu bewerten. Hierzu folgt zunächst eine Vorstellung existierender Verfahren, bevor ein weiteres entwickelt wird.

5.8.1. Übersicht existierender IMDP-Algorithmen

In der Literatur sind viele Ansätze zur Auflösung von Messabhängigkeiten bekannt, wobei einige im Abschnitt 5.5 bzw. im vorherigen Kapitel exem-

plarisch vorgestellt worden sind. Die Zusammenfassung zeigt unterschiedlichste Methoden:

- Für jeden Sensor wird ein eigener Schätzer mit eigenem Belegungsgitter verwendet (vgl. [Elfes 1989b]).
- Für jede Zelle werden die Messwinkel und -abstände gespeichert und spätere Messungen aus ähnlichen Positionen werden nicht gewertet (vgl. [Konolige 1997]).
- Für jede Zelle werden dessen Detektionseigenschaften geschätzt und dynamisch in die Evidenzmasse gewichtet (vgl. [Konolige 1997]).
- Die Evidenzmassen werden pro Messung bzw. Messzeitpunkt mittels Maschinellen Lernens bestimmt (vgl. [Thrun 1998], [van Dam 1996] und [Yu 2007]).
- Ältere Messungen werden in der Massebestimmung berücksichtigt (vgl. [Yu 2007]).
- Mittels zyklischen Markov-Ketten werden persistente Nichtdetektionen bzw. persistente Fehlziele explizit modelliert (vgl. Abschnitt 4.4.5).

Einige Verfahren lassen sich wohl recht einfach auf den vorgestellten probabilistischen Fusionsalgorithmus übertragen: So könnte wie in [Elfes 1989b] pro Sensortyp ein separater Ringpufferspeicher verwendet werden, welcher praktisch ein neues Belegungsgitter darstellt. Auch ist die Erstellung eines Bitfeldes nach [Konolige 1997] zur Dokumentation von Messwinkel und Messabstand leicht übertragbar, sodass keine doppelten Messungen von ähnlichen Positionen in den Ringpufferspeicher der Zelle eingebracht werden müssen. Auch kann die Messhistorie in den Ringpufferspeichern rekonstruiert werden, um die zyklischen Markov-Ketten nach Abschnitt 4.4.5 zu berechnen. Für Letztere ist die Festlegung einer diskreten Assoziationsschwelle notwendig, was exemplarisch im folgenden Abschnitt behandelt wird.

Die anderen vorgestellten Verfahren greifen jedoch so tief in den Kern der inversen Sensormodelle ein, dass eine approximative ICDP-Konformität mit dem Abfragealgorithmus $\beta(X)$ nicht mehr zwingend gegeben ist.

Bei der Übertragung der Verfahren nach [Elfes 1989b] und [Konolige 1997] ist mit weiteren Speicherkosten zu rechnen, da zusätzliche Daten für die Abhängigkeitsbeschreibung vorgehalten werden müssen. Jedoch

enthalten die Ringpufferspeicher bereits Informationen, die die Auflösung bestimmter Messabhängigkeiten ermöglichen: So lässt sich aus den Kennungen insbesondere bestimmen, wie häufig eine Zelle von welchem Sensor gemessen worden ist. Hierdurch können beispielsweise wiederholte Messungen eines einzelnen Sensors erkannt und geringer gewichtet werden.

Im Folgenden soll deshalb ein Algorithmus vorgestellt werden, der ausschließlich aus den bereits vorhandenen Daten versucht, Messabhängigkeiten aufzulösen.

5.8.2. Entwicklung eines IMDP-Algorithmus

Der im vorherigen Abschnitt 5.7 vorgestellte probabilistische Belegungsgitteralgorithmus ermöglicht bereits die statische Gewichtung von Messungen durch einen Faktor τ in den einzelnen inversen Sensormodellen. Jedes dieser Modelle besteht aus dem Produkt von τ und der Belegungs- bzw. Freiwahrscheinlichkeit, die sich aus der Unsicherheit von Position bzw. Sensorreichweite ergibt.

Im Fusionsalgorithmus $\xi(C,M)$ erfolgt diese Gewichtung direkt bei der Einbringung der Evidenz in den Ringpuffer. Es ist jedoch auch denkbar, diese erst nachgelagert im Abfragealgorithmus $\beta(X)$ dynamisch durchzuführen. In diesem Fall wird im Fusionsalgorithmus für alle Messungen τ auf den neutralen Faktor 1,0 gesetzt.

Wie bei Elfes sollen im ersten Schritt die Messreihen einzelner Sensoren separat ausgewertet werden. Die Abhängigkeiten zwischen den Schätzungen unterschiedlicher Sensoren werden später untersucht. Analog zum ICDP-Algorithmus werden Belegt- und Freimessungen unterschiedlich behandelt.

Erzeugung einer Belegtevidenz aus Belegtmessungen

Bei der Auswertung von Belegtmessungen soll der Umstand genutzt werden, dass eine örtliche Assoziation von Messungen möglich ist: So kann eine Messung mit einer anderen assoziiert werden, wenn beide eine bestimmte minimale Evidenz K in eine Zelle ablegen. Es wird dann davon ausgegangen, dass diese sich auf das gleiche Objekt beziehen und somit zu einem großen Teil abhängig sind.

Die mathematische Darstellung dazu ist wie folgt: Sei Z das Belegungsgitter in Verbunddarstellung nach Gleichung 5.56 eingeschränkt durch die Relation R nach Gleichung 5.57. Eine Belegtmessung eines Sensors mit der Kennung a wird mit einer anderen Messung mit der Kennung b des

gleichen Sensors in einer Zelle assoziiert, wenn folgende Relation S' erfüllt wird:

$$
\begin{aligned}
a\,S'\,b \rightarrow\ & \exists(x,y) \in (X \times Y) \wedge \exists(e_1 \in E) \wedge \exists(e_2 \in E) : \\
& \big((x,y,a,e_1) \cap Z \neq \emptyset\big) \wedge \big((x,y,b,e_2) \cap Z \neq \emptyset\big) \wedge (e_1 \geq K) \wedge (e_2 \geq K)
\end{aligned} \tag{5.65}
$$

Die Relation S erweitert S' um Transitivität, um Messungen in Nachbarzellen assoziieren zu können:

$$
\begin{aligned}
a\,S\,b \rightarrow\ & \exists\{c_1, \ldots, c_n\} \subseteq ID : \\
& (a\,S'\,c_1) \wedge (c_1\,S'\,c_2) \wedge \cdots \wedge (c_{n-1}\,S'\,c_n) \wedge (b\,S'\,c_n)
\end{aligned} \tag{5.66}
$$

Die transitiv assoziierten Ringpufferspeichereinträge werden zu einer Menge $Q(Z,id)$ zusammengefasst. Hierbei wird der daraus entstandene Messcluster durch die Messung mit der kleinsten Kennungsnummer repräsentiert:

$$
Q(Z,id) = \begin{cases} \emptyset & \text{, wenn } \exists a \in I : (a < id) \wedge (a\,S\,id) \\ \forall(x,y,b,\rho) \in Z : (b\,S\,id) & \text{, sonst} \end{cases} \tag{5.67}
$$

Die so konstruierte Menge beinhaltet alle Ringpufferspeichereinträge eines Sensors, welche sich auf ein Objekt in der gegebenen Zellmenge beziehen. Diese werden im Folgenden so fusioniert, dass eine gemeinsame Einschätzung aller Messungen für das Objekt generiert wird.

Als Beispiel dazu sind vier verschiedene Messungen in Abbildung 5.27 dargestellt, welche sich teilweise überlappen. Die Messunterschiede können beispielweise durch Messungenauigkeiten oder durch Bewegung des Objektes entstehen.

Die aus den Messungen generierten Evidenzen (rote Kästchen) werden in die Ringpufferspeicher abgelegt. Aus diesen Evidenzen zusammen mit den Kennungen und einem Schwellwert K kann die Assoziationstabelle 5.3 gebildet werden. Aus dieser ergibt sich, dass alle vier Messungen transitiv assoziierbar sind, sodass $Q(Z,1)$ aus allen in Abbildung 5.27 dargestellten Ringpufferspeichereinträgen (rote Kästchen) besteht.

Da die Evidenzen der assoziierten Messungen mit neutralem Gewichtungsfaktor τ ausschließlich die Positionsunsicherheit darstellen, wird zur besseren Approximation der Evidenz eine Mittelung aller Messungen aus

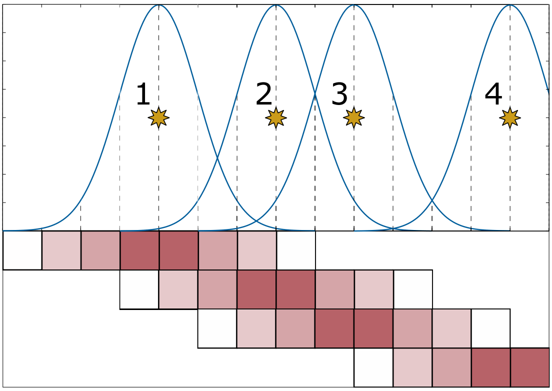


Abbildung 5.27.: Vier unterschiedliche Messungen (gelbe Sternchen) bilden in einem eindimensionalen Fahrschlauch verschiedene Messevidenzen (rote Kästchen)

ID	1	2	3	4
1	•	•		
2	•	•	•	
3		•	•	•
4			•	•

Tabelle 5.3.: Assoziationstabelle für die Messungen aus Abbildung 5.27

$Q(Z,id)$ gebildet[8]. Hierzu wird zunächst die Menge der unterschiedlichen Messkennungen eines Sensors bestimmt, die in dem Assoziationscluster $Q(Z,id)$ liegen:

$$I_{H_B}(Z,id) = \bigcup_{\forall (x,y,id,\rho) \in Q(Z,id)} id \tag{5.68}$$

Die auftretenden Evidenzen werden wie im Abschnitt 5.7 beschrieben nach Gleichung 5.58 bestimmt. Dann ergibt sich die mittlere Überlappung $\overline{\Phi}'_{H_B}$ dieser Messungen aufgrund der unsicheren Position bzw. Sensorreichweite zu:

[8] Es wird hierbei angenommen, dass die Evidenzen einer Messung nicht partiell aus den Ringpufferspeichereinträgen der Menge Q herausgefallen sind, was durch eine Modifikation der Ringpufferspeicherverwaltung erreicht werden kann.

$$\overline{\Phi}'_{H_B}(Z,id) = \begin{cases} \frac{1}{|I_{H_B}(Z,id)|} \cdot \sum\limits_{\forall id \in I_{H_B}(Z,id)} \Phi'_{H_B}(Z,id) & \text{, wenn } I_{H_B}(Z,id) \neq \emptyset \\ 0 & \text{, sonst} \end{cases} \tag{5.69}$$

Die mittlere Überlappung sei am Beispiel von Abbildung 5.27 erläutert, in der eine Zellmenge dargestellt ist, die beispielsweise einen zu untersuchenden Fahrschlauch entsprechen kann. In dieser liegen drei Messungen vollständig innerhalb des Fahrschlauchs, während die Vierte sich teilweise außerhalb befindet. Letztere Messung propagiert somit eine bestimmte Wahrscheinlichkeit, dass das punktförmige Objekt außerhalb des Fahrschlauchs liegt. Insgesamt wird diese Wahrscheinlichkeit durch die anderen drei Messungen geviertelt, sodass die fusionierte Evidenz eine hohe Wahrscheinlichkeit für eine Positionierung des Objektes innerhalb des Fahrschlauchs ergibt. Die vorgestellte Funktion erlaubt somit eine Zustandsschätzung für Objekte im Belegungsgitter bzgl. einer Zellmenge.

Neben einer Mittelung der Positionen erlauben die vorhandenen Informationen im Belegungsgitter auch die Feststellung, wie häufig ein Objekt von Messungen bestätigt worden ist. Um solche wiederholte Messungen positiv zu bewerten, kann man die Größe $|I_{H_B}(Z,id)|$ des generierten Messclusters nutzen, um den Gewichtungsfaktor τ dynamisch zu adaptieren. Je größer der Messcluster, desto asymptotisch größer sollte τ sein, bis die maximale Evidenz $\tau_{H_B,max}$ für den Sensor erreicht worden ist.

Eine mögliche asymptotische Funktion wäre beispielsweise eine Variation der sigmoiden Aktivierungsfunktion A.24 aus dem Bereich der ANNs. Diese wird durch die Konstante Σ_{H_B} modifiziert, welche angibt, wie schnell die asymptotische Näherung der maximalen Evidenz für einen spezifischen Sensor erfolgt:

$$\tau_{H_B}(Z,id) = \tau_{H_B,max} \cdot \frac{1}{1+e^{-\Sigma_{H_B} \cdot |I_{H_B}(Z,id)|}} \tag{5.70}$$

Unter der Annahme, dass die Faktoren $\overline{\Phi}'_{H_B}(Z,id)$ und $\tau_{H_B}(Z,id)$ unabhängig sind und dass unterschiedliche Objekte unabhängige Detektionseigenschaften aufweisen, lässt sich die propagierte Belegungswahrscheinlichkeit aller Sensoren wie folgt zusammenfassen:

$$\overline{\Phi}_{H_B}(Z) = 1 - \prod_{i=1}^{\infty} \left(1 - \overline{\Phi}'_{H_B}(Z,i) \cdot \tau_{H_B}(Z,i)\right) \tag{5.71}$$

Um die Unabhängigkeitsannahmen in dieser Funktion zu neutralisieren, ist es denkbar, diese mittels eines überwachten Lernverfahrens zu approxi-

mieren. Hierzu sollte für jeden Sensor eine konstante Anzahl von Elementen im Eingabevektor reserviert werden, die mit den ermittelten Objektevidenzen in aufsteigender Reihenfolge belegt werden. Um eine konstante Größe des Eingabevektors zu erreichen, werden bei einer geringeren Objektanzahl die überzähligen Eingänge auf null gesetzt, während bei zu vielen Objekten diejenigen mit der geringsten Evidenz entfallen. Der Ausgabevektor besteht hingegen aus der Belegungswahrscheinlichkeit.

Erzeugung einer Freievidenz aus Freimessungen

Analog zum probabilistischen ICDP-Algorithmus wird im Folgenden die Freiwahrscheinlichkeit für jede Zelle einzeln untersucht. Hintergrund ist auch hier die Notwendigkeit der komplementären Fusion von örtlich differierenden Freievidenzen zur Fahrschlauchschätzung.

Im Folgenden wird von einem einzelnen Sensor ausgegangen, der eine Zelle direkt oder indirekt als frei messen kann und eine entsprechende Evidenz für diese generiert. Anstatt die Messungen als unabhängig anzunehmen, wird, wie bei den Belegtmessungen auch, das Mittel der Evidenzen generiert. Zur leichteren mathematischen Darstellung wird zunächst jedoch die Menge der zu assoziierenden Messkennungen erzeugt:

$$I_{H_F}(c,id) = \bigcup_{(x,y,i,\rho)\in c} \begin{cases} i & \text{, wenn } i \text{ und } id \text{ aus der gleichen Quelle stammen} \\ \emptyset & \text{, sonst} \end{cases} \tag{5.72}$$

Wie auch bei den Belegtmessungen soll die Messung mit der kleinsten Kennung den Messcluster in einer Zelle c repräsentieren. Das arithmetische Mittel seiner Evidenzen berechnet sich dann mit:

$$\overline{\Phi}'_{H_F}(c,id) = \frac{1}{|I_{H_F}(c,id)|} \sum_{\forall(x,y,i,\rho)\in c} \begin{cases} \rho & \text{, wenn } (\min(I_{H_F}(c,id)) = id) \wedge \\ & \qquad (i \in I_{H_F}(c,id)) \\ 0 & \text{, sonst} \end{cases} \tag{5.73}$$

Auch bei Freimessungen sollen wiederholte Messungen des Sensors positiv bewertet werden. Hierzu wird analog zu den Belegtmessungen eine entsprechende asymptotische Funktion vorgeschlagen, die aus Messbestätigungen einen Gewichtungsfaktor τ ableitet. Diese lässt sich mit Σ_{H_F} für einen Sensor parametrisieren, der die Geschwindigkeit der asymptotischen Näherung an die maximale Evidenz $\tau_{H_F,max}$ angibt:

$$\tau_{H_F}(c,id) = \tau_{H_F,max} \cdot \frac{1}{1+e^{-\Sigma_{H_F} \cdot |I_{H_F}(c,id)|}} \tag{5.74}$$

Wenn die Freievidenzen unterschiedlicher Sensoren und der jeweilige Gewichtungsfaktor τ als unabhängig angenommen werden, gilt für die Freievidenz der Zelle:

$$\overline{\Phi}_{H_F}(c) = 1 - \prod_{\forall(x,y,id,\rho)\in c} (1 - \overline{\Phi}'_{H_F}(c,id) \cdot \tau_{H_F}(c,id)) \tag{5.75}$$

Wie auch bei der Belegtevidenz ist es denkbar, die Unabhängigkeitsannahmen zu neutralisieren, indem man diese Funktion mittels überwachten Lernverfahren approximiert. Der Eingangsvektor hat dabei pro Sensor so viele Eingänge, wie der Ringpufferspeicher einer Zelle an Eingängen aufweist, während der Ausgabevektor aus einer entsprechenden Massenverteilung für die Freievidenz besteht. Ebenso werden nicht belegte Eingänge auf null gesetzt, während die Evidenzen in aufsteigender Reihenfolge sortiert werden, bevor sie in den Eingabevektor geschrieben werden.

Das Zusammenfassen aller Freievidenzen einzelner Zellen erfolgt im letzten Schritt mittels pessimistischem Minimum-Operator:

$$\overline{\Phi}_{H_F}(Z) = \min_{\forall c \in Z} \left(\overline{\Phi}'_{H_F}(c) \right) \tag{5.76}$$

Die Erlernung dieser Funktion mittels überwachtem Lernverfahren ist je nach Größe des Eingabevektors mehr oder weniger problematisch. Dieser hat potenziell so viele Elemente, wie Z an Zellen maximal aufweist. Alternativ ist es denkbar, stattdessen Statistiken zur Freievidenzmenge als Eingabevektor zu nutzen. Dieser könnte dann beispielsweise Mittelwert, Varianz, größtes und kleinstes Element der Evidenzmenge enthalten.

Fusion der Belegt- und Freievidenz

Das Zusammenfassen der Belegt- und Freievidenzen zu einem Gesamtzustand kann mittels bekannter Logikalgorithmen erfolgen. Hierzu müssen die Evidenzen in Massenverteilungen umgewandelt und verknüpft werden. Als Beispiel wird im Folgenden eine die Dezert-Smarandache-Massenverteilung erzeugt:

$$\overline{\beta}(Z) = DSmT\left(\langle \overline{\Phi}_{H_F}(Z), 0, 1-\overline{\Phi}_{H_F}(Z), 0\rangle, \langle 0, \overline{\Phi}_{H_B}(Z), 1-\overline{\Phi}_{H_B}(Z), 0\rangle\right) \tag{5.77}$$

Auch hier wird die Unabhängigkeit von Belegt- und Freievidenz angenommen. Dies könnte insbesondere verletzt sein, wenn ein einzelner Sensor beide Evidenztypen generiert. Dieses ist beispielsweise bei Laserscanner und Stereokameras der Fall, welche Belegt- als auch Freibereiche aus einer einzelnen Messung generieren können.

Um die Abhängigkeiten dieser beiden Massenverteilungen aufzulösen, kann auch hier ein überwachtes Lernverfahren zur Anwendung kommen. Basiert die Zielfunktion des Lernverfahrens auf einer quadratischen Minimierung der Fehlerfunktion, so strebt diese gleichzeitig eine statistische Korrektheit an. Andernfalls kann das Ergebnis mittels einer Korrekturfunktion statistisch korrekt gestaltet werden (vgl. Abschnitt 4.6.8).

5.8.3. Zusammenfassung

In diesem Abschnitt wurden verschiedene Ansätze aus der Literatur zusammengefasst, welche versuchen, genaue Evidenzmassen für Sensoren zu generieren und Messabhängigkeiten bzw. Messakkumulationen zu vermeiden. Letzteres geschieht jedoch nur sehr rudimentär, da alle Ansätze eine rekursive Zellzustandsschätzung in einer einzelnen Massenverteilung anstreben. Letztere enthält jedoch keine Information darüber, wie dieser Zustand insgesamt erreicht worden ist, sodass Abhängigkeiten neuer Informationsquellen nicht bestimmt werden können. Wenn überhaupt werden diese Abhängigkeitsinformationen nur diskret im Zellzustand angereichert (vgl. [Konolige 1997]), was aber nur gleichartige Messakkumulationen verhindert.



Abbildung 5.28.: Innerstädtische Szenerie mit vielen Objekten. Die erzeugten Messdaten stellen Informationen mit untereinander unterschiedlichen Abhängigkeitsgraden dar.

Im Gegensatz dazu ermöglichen die Informationen in den Ringpufferspeichern der einzelnen Zellen nicht nur die Rekonstruktion der Evidenzverteilung der inversen Sensormodelle, sondern auch die approximative Ermittlung der Messhistorie für einen begrenzten Zeitraum. Letzteres kann nicht nur genutzt werden, um Messakkumulationen zu vermeiden, sondern auch um eine gefilterte Positionsschätzung zu erzeugen und die Anzahl der Messbestätigungen zu bestimmen.

Insgesamt wurde ein Algorithmus vorgestellt, der diese Aspekte bereits unter Annahmen berücksichtigt und so auch komplexere Szenarien detailliert darstellen kann, wie sie beispielsweise in Abbildung 5.28 auftreten. Sind Ground Truth-Daten verfügbar, so wurde bei einigen Funktionen vorgeschlagen, diese mittels überwachten Lernverfahren zu approximieren, sodass die Anzahl von Annahmen reduziert und der Schätzer präzisiert wird.

5.9. Vergleiche und Anwendungen

5.9.1. Vergleich mit klassischem Dempster-Shafer-Ansatz

In diesem Abschnitt wird der probabilistische Fusions- und Abfragealgorithmus aus Abschnitt 5.7.2 mit dem klassischen Dempster-Shafer Ansatz verglichen. Um die Nachvollziehbarkeit zu erleichtern, wird im Folgenden eine möglichst einfache Konfiguration vorgestellt: In dieser liefert ein einzelner Sensor ein punktförmiges Objekt mitten in einem Fahrschlauch.

Als Extraktionsalgorithmus für den klassischen Dempster-Shafer-Ansatz wird der Maximumoperator verwendet, welcher beispielsweise auch in [Kammel 2008] angewandt wird: Hierbeit repräsentiert die Zelle mit der höchsten Belegungswahrscheinlichkeit die Befahrbarkeit des gesamten abgefragten Bereichs. Als inverses Sensormodell wird das Punktmodell aus 5.7.2 verwendet, welches die entsprechenden Evidenzmassen für die einzelnen Zellen für beide Ansätze generiert.

Das verwendete Belegungsgitter hat eine Auflösung von 0,25 m $\times$ 0,25 m. Die Messungen werden mittig auf eine Zelle abgebildet, wobei die Gewichtung 0,5 beträgt und die Varianz variabel (0,1 m^2 - 0,5 m^2) ist. Der Abfragebereich ist 5 m $\times$ 5 m, wobei die Messpunkte mittig in diesem Bereich liegen. Alle Messungen werden als unabhängig angenommen. Entsprechend der Varianz liegen alle Messungen praktisch vollständig im Fahrschlauchbereich, d. h. das Integral über die Wahrscheinlichkeitsdichtefunktion der Zellen im Fahrschlauch entspricht praktisch dem verwendeten Gewichtungsfaktor.

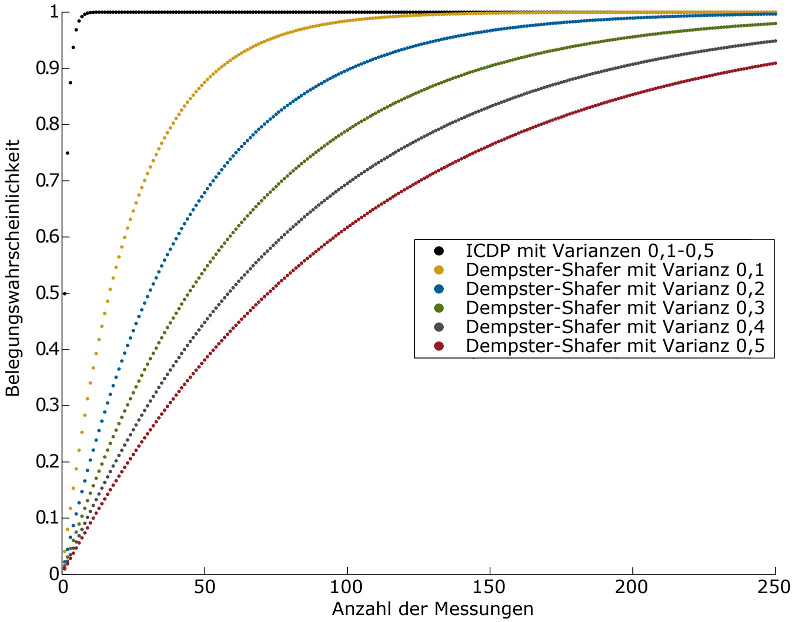


Abbildung 5.29.: Fahrschlauchbefahrbarkeit nach verschiedenen Ansätzen und Varianzen. Es wird ein punktförmiges inverses Sensormodell angenommen, wobei pro Messung der Gewichtungsfaktor 0,5 ist.

Abbildung 5.29 zeigt die verschiedenen Belegungswahrscheinlichkeiten beider Ansätze nach unterschiedlichen Messaktualisierungen. Hier sieht man deutlich, dass der klassische Ansatz viele Messungen benötigt, um eine hohe Belegungswahrscheinlichkeit zu erreichen. Hierbei gilt, dass je höher die Varianz der Messungen ist, desto mehr Messungen muss dieser Ansatz akkumulieren, um die gleiche Belegungswahrscheinlichkeit zu erreichen.

Der probabilistische Fusions- und Abfragealgorithmus hingegen erreicht sofort nach der ersten Messung den konfigurierten Gewichtungsfaktor. Nachfolgende Messungen halbieren erwartungsgemäß das Unwissen, sodass sehr schnell eine hohe Belegungswahrscheinlichkeit erreicht wird. Hierbei liefert der Algorithmus für verschiedene Varianzen praktisch das gleiche Ergebnis: winzige Unterschiede treten aufgrund nummerischer Un-

genauigkeiten auf und sind im Diagramm nicht visualisierbar. Weitere Tests zeigten, dass der ICDP-Algorithmus praktisch die gleichen Ergebnisse bei unterschiedlichen Zellauflösungen erzielt.

5.9.2. Echtzeitfähigkeit

Die vorgestellten Belegungsgitteralgorithmen können in vielfältiger Weise instanziiert, parametrisiert und beispielsweise mit einigen vorgestellten Ansätzen aus der Literatur weiter kombiniert werden. Hierbei muss die Belegungsgitterinstanz auf die eingesetzten Sensoren und die Anforderungen der jeweiligen Anwendung genau abgestimmt werden.

Eine Anwendung ist die Umfeldüberwachung im Bereich des hochautomatischen Fahrens auf Autobahnen und autobahnähnlichen Straßen bei geringen Geschwindigkeiten und in der Stillstandsphase. Beim Stillstand wird zusätzlich die Veränderung des Freibereiches überwacht: Verkleinert sich dieser, so wird davon ausgegangen, dass Fußgänger oder andere ungewöhnliche Objekte sich auf der Fahrbahn befinden und die Anfahrüberwachung wird auf den Fahrer zurückübertragen.

Das verwendete Belegungsgitter hat dabei eine einheitliche Auflösung und garantiert die Einbringung von Messungen in bis zu 100 Meter. Bis auf einige wenige sehr weit entfernte Ziele können somit alle Messungen immer in das Belegungsgitter eingetragen werden. Die genauen Parameter sind aus Tabelle 5.4 zu entnehmen.

Belegungsgitter-größe	200 m $\times$ 200 m
Zellgröße	0,25 m $\times$ 0,25 m
Zellanzahl	800 $\times$ 800 Zellen
Ringspeicher-größe	16
Speicherbedarf	ca. 91 Megabyte

Tabelle 5.4.: Verwendetes Belegungsgitter

Um das Belegungsgitter mit Evidenzen zu füllen, werden Radar-, Ultraschall- und Stereokamerasensoren eingesetzt, welche verschiedene Objekt- und Freibereichstypen generieren. Die von diesen Sensoren generierten Messungen werden durch die beschriebenen inversen Sensormo-

delle[9] in das Belegungsgitter eingetragen.

Neben den Fusionsalgorithmen greift auch der beschriebene Abfragealgorithmus auf das Belegungsgitter zu: Hierzu wird mit der in Kapitel 3 beschriebenen Fahrbahnfusion eine Fahrschlauchhypothese in Form eines Polygons generiert, welches genutzt wird, um die sich im Fahrschlauch befindlichen Zellen zu bestimmen. Der Abfragealgorithmus wird dabei inkrementell in Meterschritten in bis zu 20 Meter Entfernung ausgeführt, sodass bis zu 20 Massenverteilungen generiert werden.

Sowohl Sensoren als auch der Abfragealgorithmus besitzen eine feste Zykluszeit: Die genauen Parameter dazu sind in Tabelle 5.5 dokumentiert.

Sensor	Objekte	Messfrequenz
Stereo-kamera	bis zu 32 Rechteckobjekte und ein Freibereich	16,7 Hz
4 Ultra-schall-sensoren	je ein Freibereich und je bis zu einem radialen Objekt	je 10 Hz
Radar-sensor	bis zu 32 Punktobjekte	12,5 Hz
Bereichs-abfrage	Ein inkrementell abgefragter Fahrschlauch (ca. 1300 Zellen)	10 Hz

Tabelle 5.5.: Sensoren und Abfragealgorithmen, die das Belegungsgitter verwenden

Für das Ausgabeprotokoll werden nur wenige Massenverteilungen aufgrund vorher definierter Schwellwerte herausgegeben. Diese zeigen an, bis wann der Fahrschlauch sehr wahrscheinlich frei ist, ab wann die Schätzung unsicher ist und ab wann er sehr sicher belegt ist.

Für eine quantitative Bewertung der Algorithmen hinsichtlich Genauigkeit sind nicht nur spezifische Anwendungen notwendig, sondern zusätzlich auch Ground Truth-Daten. Letztere konnten für den Versuchsträger

[9] Die Ultraschallsensoren liefern radiale Abstandsinformationen. Das zugehörige inverse Sensormodell wurde in diesem Kapitel nicht beschrieben, da es sehr dem Punktmodell ähnelt. Die Belegungsevidenz der einzelnen Messung wird bei diesem auf einen Sektorabschnitt verteilt.

nicht generiert werden, sodass nur eine qualitative Bewertung aus dem Testbetrieb möglich ist. Hier zeigte das Belegungsgitter eine sehr hohe Zuverlässigkeit sowohl auf verschiedenen Testgeländen als auch im öffentlichen Straßenverkehr. So konnte bei Hunderten von Testkilometern keine Fehlfunktion des Belegungsgitters innerhalb der Anwendungsspezifikation festgestellt werden.

Auf der Zielplattform zeigte es auch eine gerine Latenz: So unterschreitet die Abfrage des Fahrschlauchs und die Generierung des CAN-Ausgabeprotokolls stets 1 ms bei allen getesteten Messdatensätzen. Ähnliches gilt für die Einbringung von Messungen mithilfe der inversen Sensormodelle.

Die zusätzliche Systembelastung auf der Zielplattform[10] ist gering: So benötigt die Sensordatenfusion wie im Kapitel 3 beschrieben ohne Belegungsgitter ca. 10,0 % an Systemleistung, während ca. 17,5 % mit Belegungsgitter benötigt wird. Insgesamt konnte der verwendeten Belegungsgitterinstanz eine ausreichende Echtzeitfähigkeit auf dieser Zielplattform hinsichtlich der Anforderungen des hochautomatischen Fahrens bescheinigt werden.

5.10. Zusammenfassung

Dieses Kapitel hat sich mit einer genauen und statistisch genauen Umgebungsschätzung mittels Belegungsgitter beschäftigt. Aus den Anforderungen für eine Sensordatenfusion wurde abgeleitet, dass eine solche Schätzung nicht nur für eine einzelne Zelle ermöglicht werden muss, sondern dass auch eine kombinierte Schätzung für eine Menge von Zellen unterstützt werden soll. Auch müssen möglichst viele Abhängigkeiten zwischen Messungen erkannt und beseitigt werden, um eine hohe Genauigkeit zu gewährleisten. Hierzu wurden verschiedene bekannte Ansätze untersucht, welche teilweise Messabhängigkeiten rudimentär behandeln, jedoch kaum Abhängigkeiten zwischen Zellen.

Diese beiden Probleme wurden formal als ICDP und IMDP definiert und es wurden Algorithmen gesucht, die diese beiden Probleme lösen. Leider zeigte es sich an einem unentscheidbaren Beispiel, dass konventionelle Methoden zumindest das ICDP nicht lösen können. Diese basieren auf der rekursiven Aktualisierung einer einzelnen Massenverteilung eines Zellzustandes, wobei jedoch viele Metainformation verloren gehen.

[10] Die verwendete Plattform besteht aus einem CompactPCI-Computer ausgerüstet mit einem bis zu 2,2 GHz getakteten x86-Doppelkernprozessor und 4 GB Arbeitsspeicher

Um diesem zu begegnen, wurde eine neue Umgebungsrepräsentation von Belegungsgittern vorgestellt, die sich stark von konventionellen Ansätzen unterscheidet: Hierzu wird das Belegungsgitter als Datenstruktur betrachtet, welche die diskretisierten Ergebnisse der einzelnen inversen Sensormodelle möglichst verlustfrei verwalten soll. Hierbei ist ein deutlich geringerer Informationsverlust zu verzeichnen, was beim Entwurf von Abfragealgorithmen genutzt wurde. Diese erkennen die Abhängigkeiten und Unabhängigkeiten bzgl. anderer Zellzustände und Messungen approximativ und berücksichtigen dieses in ihrem Schätzer.

Dieser Schätzer basiert darauf, alle Zellzustände so zu kombinieren, dass diese das gleiche Ergebnis liefern, als wenn eine einzelne Zelle betrachtet wird, die den gleichen Bereich abdeckt. Als Ergebnis erhält man einen Algorithmus, der innerhalb der Diskretisierungsgrenzen nicht nur approximativ unabhängig von der Zellauflösung ist, sondern auch eine statistisch genauere Umfeldschätzung ermöglicht, was insgesamt an einem Beispiel gezeigt wurde.

Da beim Abfragealgorithmus genau bestimmt werden kann, welche Sensoren wie häufig mit welcher Bereichsüberlappung den abgefragten Bereich detektiert haben, ist es später möglich, Abhängigkeiten zwischen Messungen besser aufzulösen. So filtert der vorgeschlagene Abfragealgorithmus nicht nur die Positionsinformation der Messungen, sondern gewichtet diese noch aufgrund von Messbestätigungen. Bei vorliegenden Ground Truth-Daten ist es zudem möglich, bestimmte Teilfunktionen des Schätzers genauer anzunähern und diesen insgesamt statistisch genauer zu gestalten.

Die vorgeschlagenen Algorithmen haben bei guter Parametrisierung nicht nur das Potenzial, die Genauigkeit der Umfeldschätzung zu erhöhen, sondern zeichnen sich insgesamt auch durch eine geringe Latenz aus, was an einer Beispielapplikation gezeigt wurde.

Kapitel 6.

Zusammenfassung und Ausblick

6.1. Zusammenfassung

Viele aktuelle, aber auch zukünftige Fahrerassistenzsysteme benötigen genaue, robuste und umfangreiche Umfelddaten in Echtzeit. Diese Dissertation widmete sich der Generierung eines solchen Umfeldmodells, wobei die komplette Signalkette von Sensormessungen bis zur Fahrerassistenzapplikation betrachtet worden ist. Insgesamt konnten diverse Ungenauigkeits- und Fehlerquellen sowohl bei heutigen Sensortechnologien als auch in bekannten Schätzalgorithmen identifiziert werden, wobei jedoch kein Anspruch auf eine vollständige Aufzählung aller möglichen Einflüsse besteht. Hierbei wurden die Sensortechnologien als gegeben angesehen, sodass Schätzalgorithmen Rücksicht auf deren Eigenschaften nehmen müssen und nicht umgekehrt.

Bei der Untersuchung bekannter Schätzer zeigte sich jedoch, dass diese meist bestimmte Annahmen an Eingangsgrößen machen, welche in der Realität meist nicht erfüllt werden: Sowohl bei bekannten Objektverfolgungs- als auch Belegungsgitteransätzen sollte jede als Zufallsvariable betrachtete Messung vollständig unabhängig von anderen Messungen sein, womit Abhängigkeiten innerhalb von Messreihen unberücksichtigt bleiben. Auch sollten diese Zufallsvariablen Normalverteilungen einnehmen, was jedoch bei einigen Sensoren prinzipbedingt nicht erreicht wird. Hierbei sind insbesondere Sensoren zu nennen, die diskrete Messschritte aufweisen, wie beispielsweise Laserscanner und Kamerasensoren. Bei der Auftretensrate von Scheinzielen verlangen Existenzschätzer meist eine Gleichverteilung, sodass systematische Einflüsse aus der Umwelt ignoriert werden.

Viele Schätzer weisen zudem Modellschwächen auf, wobei vorhandene Informationsquellen unberücksichtigt bleiben oder verloren gehen: So nutzen viele Signifikanzschätzer nur eine kleine Attributbasis, während bekannte Belegungsgitteransätze keine Abhängigkeiten zwischen Zellen dokumentieren und die Extrahierung von Regelgrößen selbst bei idealen Ein-

gangsdaten mit hohen Ungenauigkeiten behaftet sein kann.

Insgesamt widmete sich diese Dissertation mit der realitätsnäheren Gestaltung solcher Schätzalgorithmen: Hierzu wurden Annahmen und zugrunde liegende Modelle hinterfragt, ob sie im Kontext von aktuellen Sensortechnologien und auftretenden Objektkonstellationen das Umfeld genau darstellen können. Bei einigen Klassen von Schätzern wurden neue Ansätze vorgeschlagen, die insbesondere eine höhere Präzision gegenüber bekannten Ansätzen versprechen und weniger Annahmen an Eingangsdaten machen.

Zu den neu vorgestellten Ansätzen gehört eine Schätzung, ob ein Objekt sowohl als existent als auch signifikant anzusehen ist und somit vom Fahrerassistenzsystem berücksichtigt werden muss. Im Gegensatz zu bekannten Ansätzen wird hierbei nicht versucht, Evidenzen verschiedenster Quellen zu gewichten und mithilfe probabilistischer Logik- und Statistikalgorithmen zu akkumulieren, da dieses Vorgehen eine genaue Abhängigkeits- und Unsicherheitsanalyse aller Quellen zur Folge haben würde.

Stattdessen wird die Bestimmung der kombinierten Existenz- und Signifikanzwahrscheinlichkeit als Klassifikationsproblem aufgefasst, wobei ein entsprechender probabilistischer Klassifikator durch ein überwachtes Lernverfahren angenähert wurde. Dieser berücksichtigt automatisch alle in den Lerndaten enthaltenen Abhängigkeiten und Unsicherheiten, die beispielsweise durch Sensortechnologien, Umwelteinflüsse, Signalvorverarbeitung und Schwächen in vorgelagerten Schätzern entstehen. Er weist insgesamt eine hohe Präzision und Robustheit auf und ist echtzeitfähig.

Neben der Umfeldrepräsentation durch Objektlisten und Objektverfolgungsalgorithmen ist als weiterer Schwerpunkt die modellärmere Repräsentation mithilfe von Belegungsgittern behandelt worden. Bei der Verbesserung dieser Algorithmenklasse stand die Beseitigung von Schwächen klassischer Ansätze im Vordergrund: So berücksichtigen diese kaum Abhängigkeiten zwischen Zellinformationen und Messreihen. Diese Abhängigkeiten wurden formal als „Inter-Cell Dependency Problem“ bzw. „Inter-Measurement Dependency Problem“ definiert, sodass formale Anforderungen an Belegungsgitteralgorithmen gestellt werden konnten.

Um diesen Anforderungen zu begegnen, wurde ein neuer Belegungsgitteransatz vorgestellt: Dieser beruht darauf, das Belegungsgitter nicht als Ansammlung einzelner unabhängiger Zellen zu betrachten, sondern als Datenstruktur, welche einen möglichst verlustfreien Zugriff auf die Messhistorie in Form diskretisierter Ergebnisse inverser Sensormodelle erlaubt. Als Ergebnis konnte zu solch einem Belegungsgitter ein entsprechender Abfragealgorithmus entwickelt werden, welcher viel mehr Informationen berücksichtigt: Er kann nicht nur die Abhängigkeiten zwischen Zellen und

Messreihen besser auflösen, sondern erlaubt sogar eine Positionsfilterung.

Obwohl auch Schwächen in bekannten dynamischen Zustandsschätzern identifiziert worden sind, wurde keine direkte Verbesserung dieser Algorithmenklasse vorgeschlagen. Dies ist einerseits damit begründet, dass Kalman- oder Partikelfilter unter Modellannahmen bereits sehr gute oder sogar optimale Zustandsschätzer darstellen, andererseits sind solche Schätzer bereits stark im wissenschaftlichen Fokus gewesen, sodass zu den diversen bereits existierenden Varianten wohl nur marginale Verbesserungen erreichbar sind. Die Ungenauigkeiten dieser Filter wurden jedoch indirekt in nachfolgenden Schätzern (Relevanz, Überfahrbarkeit) berücksichtigt, welche auf überwachten Lernverfahren basieren.

Die neu entwickelten Ansätze wurden zusammen mit kalmanfilterbasierten Schätzern in eine Sensordatenfusion integriert, welche zusammen ein umfassendes Umfeldmodell repräsentieren. Die Softwarearchitektur wurde dabei so entworfen, dass hohe Modularitätsanforderungen erfüllt werden: Sie erlaubt die einfache Erstellung von kaskadierten Schätzalgorithmen und unterstützt verschiedenste Sensortechnologien und Fahrerassistenzapplikationen. Eine Einbettung in unterschiedliche Umgebungen zwecks Simulations- und Testzwecken ist durch schichtenbasierte Architekturmuster und durch beliebige Zeitquellen möglich.

Insgesamt konnte eine Sensordatenfusion geschaffen werden, die hohe Anforderungen an Präzision, Robustheit, Echtzeitfähigkeit und Modularität erfüllen kann. Sie wurde bereits auf acht Versuchsträger mit unterschiedlichen Sensortechnologien portiert, wo sie unterschiedlichste Regelgrößen in hoher Qualität für diverse Fahrerassistenzapplikationen generiert.

6.2. Ausblick

Viele heutige Fahrerassistenzsysteme basieren auf dem Konzept, dass ein Sensor genau eine Fahrerassistenzapplikation bedient. Es existieren jedoch bereits einige Systeme, die die Daten mehrerer Umfeldsensoren in einer Sensordatenfusion nutzen, um ein qualitativ besseres Umfeld zu schätzen. Dieser Trend wird sich mit der zunehmenden Anzahl an Fahrerassistenzapplikationen und deren höheren Anforderungen verstärken.

Eine Herausforderung bleibt es, aus der sich abzeichnenden Flut von Sensordaten und anderer Informationsquellen ein möglichst genaues Umfeldmodell zu erzeugen, wobei Messungenauigkeiten, systematische Fehler, Umwelteinflüsse und sonstige Abhängigkeiten stets berücksichtigt werden müssen. Diese Faktoren scheinen dabei exponentiell mit der Sensoranzahl und den zu schätzenden Entitäten in den einzelnen Domänen zu

steigen, sodass eine genaue und robuste Umfeldschätzung bereits heute, aber insbesondere auch in Zukunft einen hohen Forschungs- und Entwicklungsaufwand nach sich zieht.

Diese Arbeit hat hierzu einen kleinen Baustein geliefert, indem neue genauere Schätzmethoden zur Relevanz von Umfeldobjekten und zur Überfahrbarkeit von Arealen vorgestellt worden sind. Hierbei wurde eine möglichst unabhängige Darstellung von bestimmten Sensortechnologien und Fahrerassistenzapplikationen beschrieben, sodass diese Methoden universell anwendbar sind und auch gut mit einer zunehmenden Anzahl von Sensoren skalieren.

Mögliche Fortschritte im Bereich der Sensordatenfusion stoßen jedoch irgendwann an Grenzen: Die Schätzbasis besteht schließlich aus einer endlichen Menge nicht idealer Messdaten, aus denen man selbst mit perfekten Modellen nur im endlichen Umfang endlich genaue Informationen extrahieren kann. Um qualitativ hochwertige Umfeldmodelle zu erstellen, sind deshalb gemeinsame Fortschritte sowohl bei Sensortechnologien und zugehöriger Signalvorverarbeitung als auch im Bereich der Hypothesengenerierung und Sensordatenfusion wichtig.



(a) Zwei Fussgänger, welche durch Bewegung und Perspektive kein einheitliches Muster bilden und einem unregelmäßigen Vorder- und Hintergrund ausgesetzt sind

(b) Bild von maschinell verzerrten Zahlen auf unregelmäßigem Hintergrund, welches eine automatische Unterscheidung von Menschen und Maschinen ermöglicht (Turing-Test)

Abbildung 6.1.: Verschiedene Muster, die für Maschinen schwer wahrnehmbar sind

Zwar sind Fortschritte in der Umfeldwahrnehmung absehbar, jedoch wird in naher und mittlerer Zukunft wohl kaum die allgemeine Leistungsfähigkeit eines durchschnittlichen Fahrers in allen Verkehrsszenarien erreicht werden. Dies liegt vor allem in der für Menschen optimierten Verkehrsinfrastruktur, die genau dessen Stärken berücksichtigt: Dieser nimmt sehr viele

analoge Informationen wahr, wie Farben, Licht, Symbole, Markierungen und akustische Signale, die jeweils die hochgradig gut ausgeprägte Mustererkennung von Menschen anspricht. Mustererkennung ist jedoch gerade für Maschinen eine schwere Aufgabe, zumal die Umgebung mit vielen anderen Störmustern und Interferenzen behaftet ist. Dieser Umstand wird in anderen Bereichen sogar dazu genutzt, Menschen von maschinell simulierten Nutzern automatisch zu unterscheiden (s. Abbildung 6.1).

Letztendlich muss man feststellen, dass die Verkehrsinfrastruktur von Menschen für Menschen gemacht worden ist, und der Versuch diese genauso gut maschinell wahrzunehmen und zu interpretieren dem Versuch gleicht, Maschinen teilweise zu vermenschlichen. Sollte dieses Vorhaben irgendwann gelingen, sind wohl intelligente Systeme im Sinne eines alternativen Turing-Tests geschaffen worden.

Einige wenige Domänen im Verkehrsbereich weisen zwar exemplarisch bereits eine für Maschinen geeignete Umgebung auf: Hierbei sind beispielsweise autonom fahrende U-Bahnen oder Fahrstühle zu nennen, welche hauptsächlich in begrenzten und wohldefinierten Umgebungen digital gesteuert bzw. geregelt werden. Ein weltweiter Umbau der kompletten Verkehrsinfrastruktur hin zu einer von Maschinen sehr gut wahrnehmbaren Umgebung ist jedoch aus Kostengründen unabsehbar. Deshalb wird es wohl in naher Zukunft kein autonomes Fahrzeug geben, welches in allen Domänen eine vergleichbar leistungsfähige Fahrzeugführung aufweist, wie der Mensch.

Dennoch können punktuelle maschinelle Stärken genutzt werden, um die Auswirkungen bestimmter menschlicher Schwächen bei der Fahraufgabe zu verringern: Im Gegensatz zum Menschen werden Maschinen nicht durch Übermüdung, Gesundheitszustand, Emotionen oder Unachtsamkeit negativ beeinflusst. Auch eine möglicherweise hohe Reaktionsfähigkeit auf plötzlich auftretende Ereignisse ist von Vorteil. Ziel der nahen und mittleren Entwicklung muss es deshalb sein, die Kooperation von Fahrerassistenzsystemen und Fahrern zu perfektionieren. Dieses wird im Rahmen einer evolutionären Entwicklung auch gelingen, sodass Fahrern schrittweise eine sicherere, komfortablere und effizientere Fahrzeugführung ermöglicht wird. Dieser Fortschritt wird sich beispielsweise auch in der Reduktion von Verkehrsunfallzahlen ausdrücken. Eine autonome und sichere Fahrzeugführung in allen Domänen unter den heutigen verkehrlichen Bedingungen bleibt vorerst jedoch eine Zukunftsvision.

Anhang A.

A.1. Begriffsdefinitionen

A.1.1. Echtzeitfähigkeit

In vielen Systemen besteht die Notwendigkeit, dass bestimmte Teilsysteme innerhalb einer gewissen Zeitspanne eine Messung bzw. Berechnung ausführen, damit das Gesamtsystem funktionieren kann. Im Bereich der Fahrerassistenzsysteme sind beispielsweise solche Systeme betroffen, die eine kontinuierliche Querführung erlauben. Werden Messdaten zu spät erfasst, können hohe Osszilationen in der Querablage auftreten. Für solche Systeme werden kontinuierlich aktuelle Messdaten benötigt, welche ein definiertes Alter nicht überschreiten.

Definition (Hart Echtzeitfähigkeit)**.**
Ein (Teil-)System ist genau dann hart Echtzeitfähigkeit, wenn es eine bestimmte vorgegebene Reaktionszeit nicht überschritten wird.

Um eine harte Echtzeitfähigkeit zu erreichen, müssen zum einen Algorithmen eingesetzt werden, welche immer voraussagbar terminieren. Zum anderen muss entsprechende Hardware verwendet werden, welche die Abarbeitung der Algorithmen in der vorgegebenen Zeitspanne immer gewährleistet. Falls der Terminierungszeitpunkt für einen Algorithmus variabel ist, muss der ungünstigste Fall berücksichtigt werden.

Weitere abhängige Anforderungen an ein hartes Echtzeitsystem und die Unterscheidung zu weichen Echtzeitsysteme ist in [Kopetz 2011, S. 6ff] zu finden.

A.1.2. Statistisch korrekte Schätzer

Die Eigenschaft „Statistisch korrekt“ erlangt ein Schätzer, wenn er immer echte Wahrscheinlichkeiten als Gütemaß liefert und kein Gütemaß unbekannter Skalierung.

Hierbei unterscheiden wir zwei Klassen von Schätzern: Die Klassifikatoren und Zustandsschätzer. Während ein probabilistischer Klassifikator eine

Wahrscheinlichkeit für eine Klassenzugehörigkeit ermittelt, liefert ein probabilistischer Zustandsschätzer zusätzlich zur Schätzgröße beispielsweise eine Varianz als Gütemaß. Statistisch korrekte Klassifikatoren werden in [DeGroot 1982] behandelt[1].

Definition (Statistisch korrekte Klassifikatoren)**.**
Sei $\kappa(x)$ *ein probabilistischer Klassifikator, der für einen Merkmalsvektor* x *die Wahrscheinlichkeit schätzt, dass er einer bestimmten Klasse angehört. Sei* $X = \{x_1, \dots\}$ *eine Menge von Merkmalsvektoren und* $Y = \{y_1, \dots\}$ *die binäre Beschreibung der Klassenzugehörigkeit (*$y_i \in \{0,1\}$*). Ferner sei* $\widetilde{X} \subset X$ *mit* $(\forall \widetilde{x}_i \in \widetilde{X})|(\varepsilon_1 < \kappa(\widetilde{x}_i) < \varepsilon_2)$ *und* $\widetilde{Y} \subset Y$ *die Menge der jeweils assoziierten Klassenbeschreibung.*

Ein Klassifikator ist genau dann statistisch korrekt wenn für alle $\varepsilon_1 < \varepsilon_2$ *gilt:*

$$\sum_{i=1}^{\infty} \kappa(\widetilde{x}_i) - \widetilde{y}_i = 0 \tag{A.1}$$

Falls ein Klassifikator statistisch nicht korrekt ist, so kann man diesen mittels Korrekturfunktion statistisch korrekt gestalten.

Definition (Statistisch korrekte probabilistische Zustandsschätzer)**.**
Sei $\varsigma(x)$ *ein probabilistischer Güteschätzer, der für eine Schätzgröße* x *dessen Varianz schätzt. Sei* $X = \{x_1, \dots\}$ *eine Menge von Schätzgrößen und* $Y = \{y_1, \dots\}$ *die Menge der zugehörigen Zustandsgrößen. Ferner sei* $(\widetilde{X} \subset X)$ *mit* $\forall \widetilde{x}_i \in \widetilde{X}|(\varepsilon_1 < \varsigma(\widetilde{x}_i) < \varepsilon_2)$ *und* $\widetilde{Y} \subset Y$ *die Menge der jeweils zugehörigen Zustandsgrößen.*

Ein Schätzer ist genau dann statistisch korrekt wenn für alle $\varepsilon_1 < \varepsilon_2$ *gilt:*

$$\sum_{i=1}^{\infty} \varsigma(\widetilde{x}_i) - (\widetilde{x}_i - \widetilde{y}_i)^2 = 0 \tag{A.2}$$

Analog zur Klassifikatorfunktion kann auch die Schätzgüte mithilfe einer Korrekturfunktion statistisch genau gestaltet werden[2].

[1] Der Artikel stammt aus dem Bereich der Meteorologie, da im Bereich der Robotik bzw. Fahrerassistenzsysteme keine Quelle gefunden werden konnte, die eine geschätzte Wahrscheinlichkeit aus Ground-True-Daten validiert und gegebenenfalls korrigiert. DeGrooth nennt solche Schätzer im Meteologiebereich „perfekt zuverlässige Vorhersager“. Für diese Arbeit wurde stattdessen der Begriff „statistisch korrekt“ gewählt, da „Zuverlässigkeit“ im Software- und Technikbereich bereits eine anderweitige Bedeutung hat.

[2] Zu diesem Aspekt konnte leider keine Literaturquelle gefunden werden und er wird in dieser Arbeit auch nicht weiter behandelt.

A.2. Kalmanfilter

Das Kalmanfilter [Kalman 1960] ist ein dynamischer Zustandsschätzer, der in seiner ursprünglichen Form ein lineares Dynamikmodell F_t voraussetzt. Es ist ein sog. „Prädiktor-Korrektor-Schätzer“, welcher den aktuell geschätzten Zustand $\hat{x}_{t-1|t-1}$ eines Objektes auf den Zeitpunkt t einer Messung prädiziert und diesen anschließend anhand der Messung korrigiert.

Mithilfe des geschätzten Zustandes und des Dynamikmodells können fehlende oder fehlerhafte Messungen teilweise kompensiert werden. Hierzu wird der Zustand auf den aktuellen Zeitpunkt prädiziert, was je nach Stärke des Dynamikrauschens eine mehr oder weniger genaue neue Schätzung ohne neu eingebrachte Messungen darstellt. Der Algorithmus erweist sich unter solchen Umständen als sehr robust und vermindert die Notwendigkeit schneller Messdatenverarbeitung.

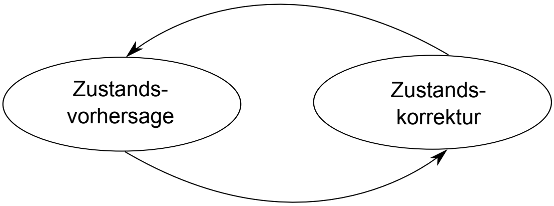


Abbildung A.1.: Ablauf rekursiver Prädiktor-Korrektor-Schätzer

Die Zustandsschätzung bildet einen rekursiven Algorithmus (s. Abbildung A.1), wobei die Prädiktion des Zustandes wie folgt aktualisiert wird:

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} \tag{A.3}$$

Neben der Zustandsprädiktion erfolgt auch die Prädiktion ihrer Unsicherheit in Form einer Kovarianzmatrix $P_{t-1|t-1}$. Da das Dynamikmodell häufig nicht perfekt ist, wird in der Regel noch ein Dynamikrauschen Q_t definiert, welches die Zustandsunsicherheit weiter erhöht:

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \tag{A.4}$$

Die maximal mögliche Korrektur des Zustandes wird durch die sog. „Innovation“ $\tilde{y}_t$ festgelegt. Um diese zu bestimmen, wird der geschätzte Zustand $\hat{x}_t$ mit der Messung z_t verglichen. Dafür wird der Schätzzustand mithilfe der Matrix H_t in den Messraum abgebildet:

$$\tilde{y}_t = z_t - H_t \hat{x}_{t|t-1} \tag{A.5}$$

Die sog. „Residualkovarianz“ S_t beschreibt die Unsicherheit der Innovation in Form ihrer Kovarianzmatrix. Sie berechnet sich aus der geschätzten Zustandskovarianz P_t und der apriori bestimmten Messkovarianz R_t folgendermaßen:

$$S_t = H_t P_{t|t-1} H_t^T + R_t \tag{A.6}$$

Aus der aktuellen Zustandsunsicherheit $P_{t|t-1}$ und der Residualkovarianz S_t kann der optimale Kalmanverstärkungsfaktor K_t für den späteren Korrekturschritt berechnet werden:

$$K_t = P_{t|t-1} H^T S_t^{-1} \tag{A.7}$$

Mit dem optimalen Kalmanverstärkungsfaktor K_t und der Innovation $\tilde{y}_t$ wird der Schätzzustand wie folgt aktualisiert:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \tilde{y}_t \tag{A.8}$$

K_t wird ebenso genutzt, um die zu der Schätzung gehörige Kovarianzmatrix P_t entsprechend zu aktualisieren:

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} \tag{A.9}$$

Das Kalmanfilter ist bewiesenermaßen ein optimaler Zustandsschätzer (vgl. Kalman [1960]), wenn sowohl das Dynamikmodell F_t als auch die Abbildung vom Systemraum in den Messraum H_t linear darstellbar sind und alle Unsicherheiten (Q_t, R_t) normalverteilt sind. Bei solchen Systemen ist die Genauigkeitsanforderung entweder durch das Kalmanfilter erfüllt oder sie ist nicht erfüllbar.

In vielen Anwendungsbereichen sind diese idealen Voraussetzungen jedoch nicht erfüllt. Hierfür wurden im Laufe der Zeit viele Varianten des Kalmanfilters entwickelt, welche versuchen, diese Einschränkung zu kompensieren.

So erlaubt beispielsweise das sog. „erweiterte Kalmanfilter“ auch die Verwendung von nichtlinearen und differenzierbaren Modellen, welche dann approximativ linearisiert werden. Hierbei werden im Kalmanfilter die Matrizen F_t und H_t durch entsprechende Jacobimatrizen ersetzt, welche als Elemente die partiellen Ableitungen der einzelnen Modellvariablen enthalten. Diese werden bei der Kovarianzprädiktion, bei der Residualkovarianzberechnung, bei der Berechnung des optimalen Kalmanverstärkungsfaktors und bei Kovarianzmatrixberechnung verwendet. Bei der Zustandsprädiktion und bei der Innovationsberechnung können die Funktionen direkt genutzt werden.

Die Zustandsschätzung ist je nach Stärke der Nichtlinearität der Modelle mehr oder weniger suboptimal. Eine weitere Verbesserung stellt das sog. „Unscented Kalman Filter“ dar. Diese Variante nutzt zwei sog. „Sigmapunkte“ pro Dimension des Zustandsraumes, welche um den Schätzzustand platziert werden. Diese Punktmenge wird mithilfe des potenziell nichtlinearen Dynamikmodells auf einen Messzeitpunkt prädiziert. Aus der transformierten Punktmenge lässt sich eine neue Wahrscheinlichkeitsdichtefunktion ableiten, welche für eine genauere Kovarianzaktualisierung genutzt wird (vgl. [Julier 1997]).

A.3. Maschinelle Lernverfahren

A.3.1. Bayessches Netz

Ein Bayessches Netz wird aus einem gerichteten azyklischen Graphen gebildet, welcher die Zusammenhänge des zugrunde liegenden Klassifikationsproblems repräsentiert. Als Beispiel für solch ein Bayessches Netz ist in Abbildung A.2 eine mögliche Struktur für einen Relevanzschätzer dargestellt. Während die Knoten des Graphen die Zufallsvariablen beschreiben, markieren die Kanten die bedingten Abhängigkeiten zwischen diesen [Pearl 1988, S. 77 ff]. Die Zufallsvariablen werden häufig in diskreter Form in den Knoten abgelegt. Bei dieser Modellierungsform müssen kontinuierliche Eingangsdaten diskretisiert werden.

Bei einem Bayesschen Netz muss zum einen eine Netzstruktur festgelegt werden, die die Abhängigkeiten des zugrunde liegenden Problems repräsentieren. Ist diese unbekannt, so kann man auf Algorithmen zurückgreifen, die aus Trainingsdaten eine mögliche Netzstruktur mittels lokaler Minimierung bestimmen. Als Beispiel sei der sog. „K2-Algorithmus“ [Cooper 1992]) genannt.

Zum anderen muss beim Bayessches Netz eine Merkmalsdiskretisierung von kontinuierlichen Eingangswerten bestimmt werden. Ist diese zu grob, so können relevante Zustände nicht mehr differenziert werden. Ist sie zu klein, kann es zu einer Überanpassung an die Trainingsdaten kommen [Uusitalo 2007].

Außerdem hat die Zahl der Diskretisierungsschritte einen großen Einfluss auf die bedingte Verbundwahrscheinlichkeitstabelle eines Netzknotens. So steigt deren Größe exponentiell zur Anzahl der Eingangsknoten n, während die Anzahl der Diskretisierungsschritte d_i die Basis bildet. Die Größe der Tabelle ist gegeben durch:

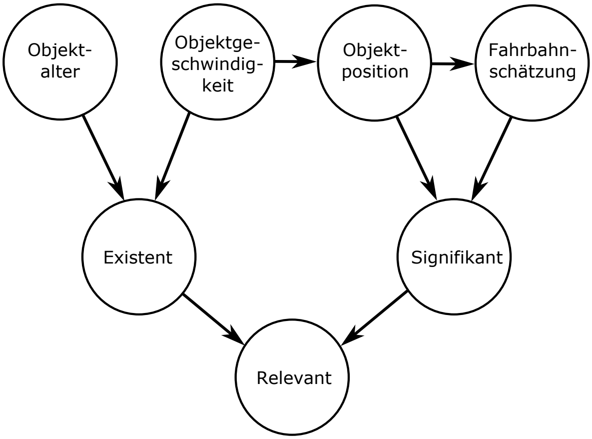


Abbildung A.2.: Mögliche vereinfachte Struktur eines Bayesschen Netzes zur Schätzung der Relevanzwahrscheinlichkeit

$$O(n) = \prod_{i=1}^{n} d_i \tag{A.10}$$

Diese Größenabschätzung bildet gleichzeitig die untere Schranke für die Komplexität der Trainingsalgorithmen. Aufgrund der potenziell hohen Anzahl von freien Variablen, die eine Verbundverteilung mit vielen Eingangsparametern nach sich zieht, ist es für die Trainingsalgorithmen eine Herausforderung, keine Überanpassung vorzunehmen.

Die Verwendung von Algorithmen zur optimalen Bestimmung von Diskretisierungsschritten ist wenig verbreitet, sodass hier in den meisten Fällen ein Experte diese vornimmt.

A.3.2. k-Nearest-Neighbour

Ein weiterer Algorithmus ist der sog. „k-Nearest-Neighbour-Algorithmus (kNN)“, welcher die k nächsten Ground Truth-Daten zurate zieht, um eine Klassifikation zu treffen [Dasarathy 1990]. Die Relevanzwahrscheinlichkeit berechnet sich aus dem Quotienten aus der Anzahl der relevanten Objekte der k nächsten Objekte und k.

Die richtige Wahl von k hat großen Einfluss auf das Klassifikationsergebnis: Ist k zu groß gewählt, kann er bestimmte Merkmale bei wenig dichten

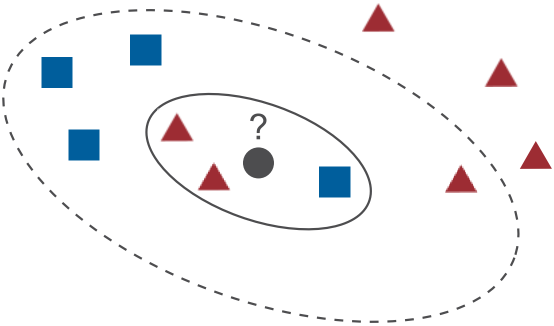


Abbildung A.3.: k-Nearest-Neighbour Klassifikator mit k=3 und k=7 und Mahalanobisabstandsmaß

Eingangsdaten nicht mehr differenzieren. Ist k zu klein gewählt, so kann eine Überanpassung erfolgen (s. Abbildung A.3). Zudem benötigt der kNN-Algorithmus eine Abstandsfunktion, um zu entscheiden, welche Objekte dem Muster am ähnlichsten sind. Hierbei wird häufig eine Mahalanobisdistanz verwendet, dessen Parametrisierung auf einer Verteilungsanalyse der Eingangsdaten beruht. Dabei ist der Klassifikator darauf angewiesen, dass die Eingangsdaten normalverteilt sind. Ist dieses nicht der Fall, kann ein Experte versuchen, diese mittels einer Abbildungsfunktion entsprechend aufzubereiten.

Der kNN-Algorithmus besitzt im Prinzip keine Lernphase, welche Trainingsdaten kondensiert und überflüssige Daten eliminiert. Er gehört somit der Kategorie der sog. „Lazy Learner"-Algorithmen an, welche die Klassifizierungsfunktion direkt auf die Trainingsdaten anwendet. Vorteilhaft ist, dass die Trainingsdatenbasis mit sehr geringem Aufwand erweitert werden kann. Insbesondere bei einer großen Trainingsdatenbasis ist es jedoch nachteilig, dass der Klassifikator einen hohen Ressourcenverbrauch in Form von Speicherplatz und je nach Implementation auch in Form von Rechenzeit beansprucht.

A.3.3. Lineare Support Vector Machine

Die zugrunde liegende Struktur einiger Klassifikationsprobleme ist möglicherweise weniger komplex, sodass diese bereits von Klassifikatoren mit einer geringen Anzahl an Freiheitsgraden ausreichend approximiert werden können. Ein solch einfacher Klassifikator ist die Hyperebene, welche

für linear separable Daten geeignet ist. Im Folgenden werden Ansätze zur Approximation einer solchen Ebene nach [Fan 2008] und [Burges 1998] vorgestellt.

Eine solche Trennebene kann hierbei durch einen Normalenvektor w und einen Versatz b dargestellt werden, sodass für alle Punkte $x_i \in X$ der Trainingsmenge X gilt:

$$w \cdot x_i - b \geq 0 \quad \text{, wenn } x_i \text{ der Klasse A angehört} \tag{A.11}$$

$$w \cdot x_i - b < 0 \quad \text{, sonst} \tag{A.12}$$

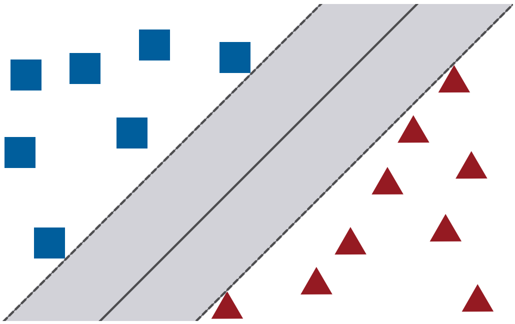


Abbildung A.4.: Der Abstand der eigentlichen Trennebene (durchgezogene Linie) und parallelen Trennebenen (gestrichelte Linien) sollte für eine robuste Klassifikation maximiert werden

Bei linear separablen Daten existieren meist unendlich viele mögliche Trennebenen. Um den Klassifikator möglichst robust zu gestalten, sollte hierbei die Trennebene gewählt werden, welche den Abstand zu den Trainingsdaten maximiert. Hierzu werden zwei weitere parallele Ebenen definiert, welche jeweils Abstand von $\frac{1}{||w||}$ zur eigentlichen Trennebene aufweisen (vgl. Abbildung A.4):

$$w \cdot x_i - b \geq 1 \quad \text{, wenn } x_i \text{ der Klasse A angehört} \tag{A.13}$$

$$w \cdot x_i - b < -1 \quad \text{, sonst} \tag{A.14}$$

Um den Abstand der Trennebene zu den parallelen Ebenen zu maximieren, ist die Minimierung von $||w||$ bzw. $||w||^2 = w \cdot w$ notwendig. Hierzu

kann auf bekannte Verfahren im Bereich der konvexen Optimierung zurückgegriffen werden. Zu bemerken dabei ist, dass meist nur wenige Punkte in den Trainingsdaten die optimale Trennebene definieren: Diese werden Stützvektoren (engl. „support vectors“) genannt, welche mittels Maschinellen Lernens bestimmt werden. Eine Trennebene kann somit auch durch eine entsprechende Stützvektormenge beschrieben werden, was dem Verfahren den Namen „Support Vector Machine (SVM)“ gegeben hat.

Die beschriebene Trennung funktioniert nur bei exakt linear trennbaren Daten, sodass ein einzelner Ausreißer diese Methode bereits unbrauchbar machen könnte. Um dieses zu vermeiden, wurden Schlupfvariablen ζ_i in den Nebenbedingungen definiert, welche Ausreißer ermöglichen:

$$w \cdot x_i - b \geq \quad 1 - \zeta_i \quad \text{, wenn } x_i \text{ der Klasse A angehört} \tag{A.15}$$

$$w \cdot x_i - b < \quad -1 + \zeta_i \quad \text{, sonst} \tag{A.16}$$

Auftretende Ausreißer werden aber auch gleichzeitig bestraft, indem die Schlupfvariablen ζ_i in der zu optimierenden Zielfunktion berücksichtigt werden. Wie stark Ausreißer gewichtet werden, wird in dem Parameter $C > 0$ festgelegt:

$$\min_{||w||,\zeta} \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \zeta_i \right\} \tag{A.17}$$

Je nach Gewichtung mittels C kann entweder eine Trennebene priorisiert werden, welche die Trainingsbeispiele möglichst zahlreich trennt oder aber eine Trennebene, die den durchschnittlichen Abstand zu den Datenpunkten maximiert.

Die lineare Trennung von Daten hat den Vorteil, dass die Bestimmung der optimalen Hyperebene im Sinne der Zielfunktion effizient möglich ist, sodass sich dieser Klassifikator auch für sehr große und hochdimensionale Datenbestände eignet. Die Zahl der freien Parameter ist dabei auf die Dimension des Klassifikationsproblems beschränkt, sodass Überanpassungen möglicherweise weniger ausgeprägt sind als bei anderen Ansätzen. Auch ist die Klassifikation selbst sehr schnell möglich, da im Wesentlichen nur das Skalarprodukt zwischen dem zu klassifizierenden Punkt und dem Normalenvektor der Hyperebene bestimmt werden muss. Echtzeitanforderungen lassen sich somit leicht umsetzen.

Nachteilig ist die möglicherweise schlechtere Leistungsfähigkeit bei nicht linear separablen Klassifikationsproblemen. Abhilfe kann hier eventuell eine Datenanalyse und eine Abbildungsfunktion schaffen, die den Klassifikationsraum geeignet transformiert. Dieses ist manuell wohl aber nur bei

niedrigdimensionalen Problemen effektiv möglich und erfordert Modellwissen.

A.3.4. Support Vector Machine mit Kernelerweiterung

Die lineare Trennung erreicht nur bei entsprechend verteilten Daten gute Klassifikationsraten, sodass diese beispielsweise keine n-XOR- bzw. Paritätsfunktion darstellen kann. Um komplexere Trennflächen zu ermöglichen, ist eine Erweiterung des Verfahrens notwendig, welche mehr Freiheitsgrade als die Hyperebene im Merkmalsraum erlaubt. Dieser Ansatz wird beispielsweise in [Burges 1998] und [Steinwart 2008] verfolgt und im Folgenden beschrieben.

Eine Möglichkeit mehr Freiheitsgrade zu erlauben, besteht in der Abbildung des Merkmalsraums auf einen euklidischen Raum höherer Dimension, sodass eine Trennebene entsprechend mehr Parameter in Form zusätzlicher Stützvektoren aufweist. Durch eine entsprechende Rückabbildung können dann allgemeinere Trennflächen im Merkmalsraum beschrieben werden.

Da innerhalb der Trainings- und Klassifikationsalgorithmen die Elemente des Merkmalsraums nur mittels Skalarprodukten verknüpft werden, ist die Abbildung einzelner Elemente in einen höherdimensionalen Raum nicht direkt notwendig. Stattdessen ist es ausreichend, eine Abbildungsfunktion für zwei Elemente des Merkmalsraums in einen Skalarproduktraum zu definieren, welcher durch einen möglichen höherdimensionalen Raum impliziert wird. Eine solche Funktion wird Kernelfunktion genannt.

Kernelfunktionen haben den Vorteil, dass ihre Dimension durch die Dimension des Merkmalsraums definiert und begrenzt ist, jedoch deutlich höher- oder gar unendlichdimensionale Abbildungsräume implizieren können, in denen ein entsprechendes Skalarprodukt definiert werden müsste. Sie erlauben somit eine indirekte, aber effiziente Skalarproduktbildung von Elementen in solchen Räumen und erweitern somit die Echtzeitfähigkeit und Genauigkeit des Klassifikators.

Kernelfunktionen können als eine Art Distanzmaß im Zustandsraum betrachtet werden: Während beim normierten euklidischen Skalarprodukt der Abstand eines Punktes zu einer Trennebene bestimmt wird, lassen sich mit anderen Kernelfunktionen komplexere Distanzmaße definieren. Gängig sind polynomielle, radiale und sigmoide Kernelfunktionen, welche durch Konstanten (p, σ, κ und δ) parametrisiert werden und teilweise aus dem Bereich der künstlichen neuronalen Netze bekannt sind:

$$K_{Poly}(x_1, x_2) = (x_1 \cdot x_2 + 1)^p \tag{A.18}$$

$$K_{Radial}(x_1, x_2) = e^{-\frac{(x_1 - x_2)^2}{2\sigma^2}} \tag{A.19}$$

$$K_{Sigmoid}(x_1, x_2) = \tanh(\kappa x_1 \cdot x_2 - \delta) \tag{A.20}$$

Während bei der linearen Trennung die Anzahl der benötigten Stützvektoren durch die Dimension des Merkmalsraums definiert ist, können bei den genannten Kernelfunktionen beliebig viele Stützvektoren verwendet werden. Hierzu gibt es verschiedene Verfahren, mit denen iterativ eine geeignete Stützvektormenge bestimmt werden kann. Für Genaueres sei auf [Burges 1998] verwiesen.

Des Weiteren existieren Ansätze, die eine gewichtete Klassifikation erlauben. Als Bewertungsgrundlage spielen hierbei die durch Kernelfunktionen ermittelten Distanzen eine wichtige Rolle: So wird beispielsweise in [Platt 1999] eine sigmoide Funktion approximiert, welche den Entfernungen zur Trennfläche d im Skalarproduktraum entsprechende Wahrscheinlichkeiten $p(d)$ zuordnet:

$$p(d) = \frac{1}{1 + e^{A \cdot d + B}} \tag{A.21}$$

Die vorgeschlagene Funktion wird jedoch nur durch die beiden Konstanten A und B parametrisiert, sodass komplexe statistische Zusammenhänge möglicherweise nicht genau dargestellt werden. Das Ergebnis kann aber zumindest als Gütemaß unbekannter Skala betrachtet werden.

Eine beispielhafte Klassifikation einer SVM mit Kernelerweiterung ist in Abbildung A.5 dargestellt, in der ein zweidimensionaler Raum in diskreten Schritten probabilistisch klassifiziert wird.

Insgesamt eignet sich die SVM mit Kernelerweiterung für komplexe Klassifikationsprobleme, wobei die Trainingsmenge begrenzt sein sollte, damit iterative Trainingsalgorithmen nicht zu viele mögliche Stützvektoren testen müssen. Eine wichtige Rolle spielt außerdem die Trennbarkeit: So können viele Ausreißer Trainingsalgorithmen verleiten, Überanpassungen durch zusätzliche Stützvektoren vorzunehmen. Dies kann zwar mithilfe des C-Parameters aus Gleichung A.17 möglicherweise abgeschwächt werden, jedoch kann dies auch zu einer schlechteren Klassifikatoradaption führen.

Auch ist es herausfordernd, eine passende Parametrisierung für die Kernelfunktion und eine geeignete Stützvektormenge zu finden. Die Parametrisierung eines Klassifikators auf Basis von SVMs sollte deshalb mithilfe

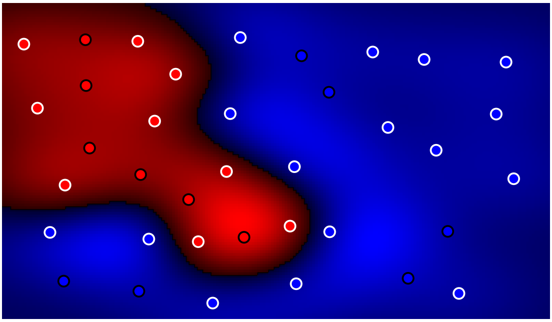


Abbildung A.5.: Probabilistische Klassifikation (roter und blauer Bereich) zweidimensionaler Merkmalsvektoren (rote und blaue Kreise) mittels radialer Basisfunktion als Kernelfunktion. Berücksichtigt werden 25 Stützvektoren (weiß umrandete Kreise).

von Kreuzvalidierungsverfahren unterstützt werden, sodass ein Parametersatz mit hoher Genauigkeit erzeugt wird. Eine mögliche Vorgehensweise wird beispielsweise in [Chang 2011] beschrieben.

A.3.5. Künstliche neuronale Netze

Künstliche neuronale Netze (engl. „Artificial Neuronal Networks (ANN)") bilden eine Klasse von Algorithmen aus dem Bereich des Maschinellen Lernens (vgl. [Haun 1998], [Borgelt 2003] und [Nissen 2003]). Ihnen allen ist gemeinsam, dass sie biologische Neuronen als Vorbild nehmen, um ein künstliches Neuron zu abstrahieren. Hierbei sammeln Neuronen über sog. „Dendriten" Energie in Form von elektrischer Aktivität und leitet diese bei Überschreitung eines Schwellwertes über sog. „Axone" weiter. Die Dendriten sind mit anderen Axonen über sog. „Synapsen" verbunden, wobei diese je nach Typ das elektrische Signal verstärken oder hemmen können.

Ein künstliches Neuron imitiert ein solches Verhalten: Es erhält Aktivitäten über verschiedene Eingänge, welche über eine sog. „Übertragungsfunktion" Σ gewichtet und zusammengefasst werden. Das Ergebnis der Übertragungsfunktion wird an eine sog. „Aktivierungsfunktion" φ weitergeleitet. Diese ermittelt die Ausgabeaktivität y des Neurons, wobei ein Schwellwertparameter θ berücksichtigt wird. In Abbildung A.6 ist die Verarbeitung exemplarisch dargestellt.

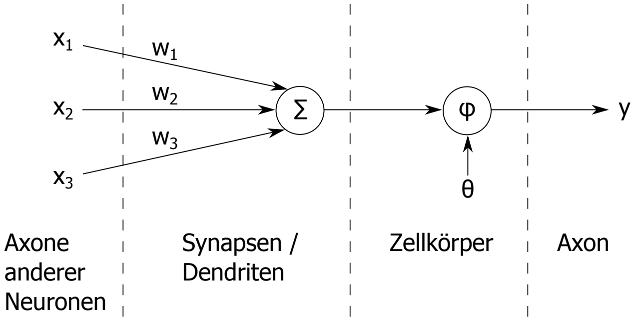


Abbildung A.6.: Abstrahiertes künstliches Neuron

Diese künstlichen Neuronen werden zu ganzen Netzen verknüpft, welche mittels Trainingsdaten in der Lage sind, Funktionen zu approximieren oder als Assoziativspeicher zu fungieren.

Im Folgenden soll ein Netztyp aus der Klasse der vorwärtsgekoppelten Netze mit überwachter Lernfunktion verwendet werden. Es handelt sich um sog. „Fehlerrückführungsnetze“ (engl. „backpropagation nets“), welche sich durch ein Schichtenlayout auszeichnen.

Hierbei werden insgesamt drei Schichttypen unterschieden. In der Wahrnehmungs- bzw. Eingabeschicht wird das zu klassifizierende Muster entgegengenommen. Dieses besteht aus so vielen Neuronen, wie Eingangsmerkmale verfügbar sind. Am anderen Ende des Netzes befindet sich die Ausgabeschicht, welche so viele Neuronen hat, wie Ausgabewerte erzeugt werden müssen. Im Fall des Relevanzklassifikators besteht dieser aus nur einem Neuron.

Dazwischen befinden sich meist ein bis zwei versteckte Schichten. Diese ermöglichen es dem Netz, beliebige Funktionsapproximationen zu betreiben [Pinkus 1999]. Jedes Neuron einer Schicht ist mit mindestens einem Neuron aus den Nachbarschichten verbunden. Häufig wird die Klasse der vollständig verbundenen Fehlerrückführungsnetze genutzt, sodass ein Neuron einer Schicht mit allen Neuronen der Nachbarschichten verbunden ist. Ein Beispiel für ein solches Netz ist in Abbildung A.7 dargestellt.

Um ein Eingabemuster von solch einem Netz klassifizieren zu lassen, werden die Eingangsmerkmale meist auf das Intervall [0, 1] oder [-1, 1] normiert. Diese bilden die Eingangswerte der Wahrnehmungsschicht.

Die Eingangswerte werden über gewichtete Neuronenverbindungen in die nächste Schicht geleitet und von den dortigen Neuronen entgegen-

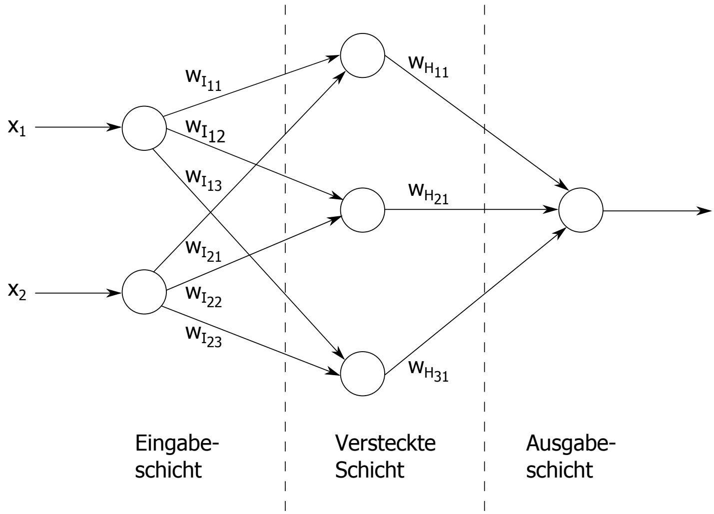


Abbildung A.7.: Vollständig verbundenes mehrschichtiges künstliches neuronales Netz

genommen. Um alle Eingänge zu kombinieren, wird eine Übertragungsfunktion verwendet. Sei $X = \{x_1, \ldots x_n\}$ die Menge der Eingangswerte und $W = \{w_1, \ldots, w_n\}$ die Menge der Verbindungsgewichte eines Neurons. Die Übertragungsfunktion $\Sigma(X,W)$ liefert als Netzeingabe eine Energie nach:

$$\Sigma(X,W) = \sum_{i=1}^{n} w_i x_i \tag{A.22}$$

Während die Eingangswerte der Wahrnehmungsschicht meist nach dem Empfang identisch weitergeleitet werden, wird in den anderen Fällen auf die Netzeingabe eine sog. „Aktivierungsfunktion" angewendet. Diese muss für das später beschriebene Lernverfahren differenzierbar sein.

Meist wird eine sog. sigmoide Aktivierungsfunktion verwendet, welche es dem künstlichen neuronalen Netz ermöglicht, nichtlineare Zusammenhänge wiederzugeben. Diese zeichnet sich durch ihren beschränkten Wertebereich, einer streng monotonen Steigung und genau einen Wendepunkt aus. Eine häufig genutzte sigmoide Aktivierungsfunktion ist ein Spezialfall der logistischen Funktion, welche in Abbildung A.8 dargestellt ist. Diese und deren partielle Ableitung ist beschrieben durch:

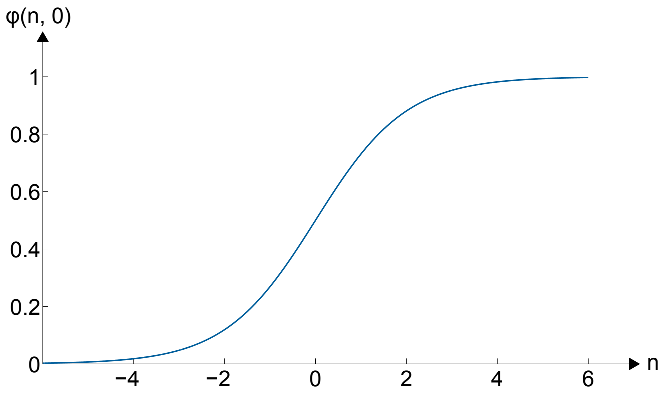


Abbildung A.8.: Sigmoide Aktivierungsfunktion

$$\varphi(n,\theta) = \frac{1}{1+e^{-(n-\theta)}} \tag{A.23}$$

$$\frac{\partial\varphi(n,\theta)}{\partial n} = \varphi(n,\theta)\cdot(1-\varphi(n,\theta)) \tag{A.24}$$

Bis auf die Ausgabeschicht wird in den anderen Schichten meist noch ein spezielles Neuron eingefügt, welches ständig aktiv ist. Dieses bewirkt eine konstante Addition auf das Ergebnis der Übertragungsfunktion mit dessen Verbindungsgewicht und ersetzt somit innerhalb der Aktivierungsfunktion den Schwellwertparameter θ. Somit müssen nur noch die Verbindungsgewichte erlernt werden.

Um eine Ausgabe zu erzeugen, wird der Eingabeschicht ein Muster vorgegeben. Schichtenweise werden die Übertragungs- und Aktivierungsfunktionen berechnet. Das Ergebnis der Aktivierungsfunktion der einzelnen Neuronen der Ausgabeschicht repräsentiert das Klassifikationsergebnis des gesamten künstlichen neuronalen Netzes. Die gesamte Parametrisierung des Netzes ist in den Verbindungsgewichten W abgelegt. Sie zu ermitteln ist das Ziel des Lernverfahrens des sog. „Fehlerrückführungs-Algorithmus“ (engl. „Backpropagation Algorithm“) [Rumelhart 1986].

Dazu werden die Gewichte im ersten Schritt mit zufälligen Werten initialisiert. Das künstliche neuronale Netz liefert jetzt eine zufällige Ausgabe auf ein Eingabemuster. Im Folgenden sollen die Gewichte so modifiziert werden, dass das künstliche neuronale Netz alle Muster möglichst fehler-

frei klassifiziert. Hierzu ist eine repräsentative Menge von zu klassifizierenden Mustern und deren gewünschte Ausgabe notwendig. Das künstliche neuronale Netz minimiert den durchschnittlichen quadratischen Fehler der Mustermenge gemäß Gleichung 4.29.

Um die Verbindungsgewichte zu bestimmen, ist zu unterscheiden, ob es sich um eine Verbindung zwischen versteckter Schicht und Ausgabeschicht oder zwischen Eingabeschicht und versteckter Schicht handelt. Im ersten Fall hat das Gewicht direkten Einfluss auf die Ausgabe, im anderen Fall kann der Einfluss nur indirekt ermittelt werden. Die Gewichtsänderung wird aus dem Fehleranteil ε berechnet, den das Neuron bewirkt hat. Dieser berechnet sich für ein Neuron n_{O_j} der Ausgabeschicht aus der Differenz zwischen Soll-Wert $\omega_{n_{O_j}}$ und Ist-Wert $v_{n_{O_j}}$ mit:

$$\varepsilon_{n_{O_i}} = \omega_{n_{O_j}} - v_{n_{O_j}} \tag{A.25}$$

Aus dem Fehlerwert $\varepsilon_{n_{O_j}}$ wird ein Korrekturwert $\delta_{n_{O_j}}$ abgeleitet, welcher später sowohl für die Berechnung der Gewichtsänderungen als auch für die Fehlerwertberechnung der Neuronen vorangegangener Schichten benötigt wird. Mittels einschrittigem Newtonverfahren wird die Funktionswertdifferenz approximiert, welche den Fehlerwert verkleinert. Hierbei wird die in Gleichung A.24 genannte Ableitung der Aktivierungsfunktion verwendet[3].

$$\delta_{n_{O_j}} = \varepsilon_{n_{O_j}} \cdot \varphi(x_{n_{O_j}}, 0) \cdot [1 - \varphi(x_{n_{O_j}}, 0)] \tag{A.26}$$

Nur ein Bruchteil von $\delta_{n_{O_j}}$ wird genutzt, um die Gewichte anzupassen. Der Anteil ist durch eine konstante Lernrate η festgelegt. Somit ändert sich das Gewicht $w_{H_{ij}}$ zwischen einem Neuron n_{H_i} der versteckten Schicht und einem Neuron n_{O_j} der Ausgabeschicht mit:

$$w_{H_{ij}} = w_{H_{ij}} + \eta \delta_{n_{O_j}} \tag{A.27}$$

Die Fehlerwertberechnung für ein Neuron n_{H_i} der vorangegangenen versteckten Schicht erfolgt indirekt: Dessen Fehlerwert $\varepsilon_{n_{H_i}}$ entspricht der mit den Neuronenverbindungen $w_{H_{ij}}$ gewichteten Summe der zuvor bestimmten Korrekturwerte $\delta_{n_{O_j}}$:

$$\varepsilon_{n_{H_i}} = \sum_{j=1}^{n} (w_{I_{ji}} \delta_{n_{O_i}}) \tag{A.28}$$

Aus dem Fehlerwert $\varepsilon_{n_{H_i}}$ wird analog zu Gleichung A.26 wieder ein Korrekturwert $\delta_{n_{H_i}}$ abgeleitet und mit dessen Hilfe analog zu Gleichung A.27

[3] Wie beschrieben, wurde der Schwellwert θ in ein zusätzliches Verbindungsgewicht ausgelagert, sodass dieser Parameter in der Aktivierungsfunktion null ergibt

die Gewichte der Neuronenverbindungen zwischen Eingabeschicht und versteckter Schicht angepasst.

Die Methode der Gewichtsanpassung stellt sich insgesamt als Gradientenabstiegsverfahren im hochdimensionalen Fehlerfeld dar. Über die Ableitungsfunktion des Fehlers wird die Bewegungsrichtung im Feld genutzt, die den steilsten Abstieg verspricht. Die begrenzte Lernrate η sorgt dafür, dass ein nahes lokales Minimum nicht übersprungen wird.

Die Anpassung der Gewichte erfolgt iterativ, wobei als Abbruchkriterium die Entwicklung des durchschnittlichen quadratischen Fehlers genutzt wird. Sollte dieser einen stagnierenden oder gar negativen Trend aufweisen, so ist das Abbruchkriterium erfüllt. Um eine Überanpassung zu vermeiden, sollte der quadratische Fehler jedoch für eine separate Testmenge und nicht für die Trainingsdaten bestimmt werden.

Literaturverzeichnis

[Ahrholdt 2006] AHRHOLDT, M. ; BEUTNER, A.: Advanced Driver Assistance for Trucks by Lane Observation. In: *Proceedings of the 3. International Workshop on Intelligent Transportation*, Hamburg, 2006

[Alpaydin 2004] ALPAYDIN, E.: *Introduction to machine learning*. The MIT Press, 2004

[Altendorfer 2010] ALTENDORFER, R. ; MATZKA, S.: A Confidence Measure for Vehicle Tracking based on a Generalization of Bayes Estimation. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, San Diego, USA, 2010, S. 766–772

[Bar-Shalom 1988] BAR-SHALOM, Y. ; FORTMANN, T. E.: *Tracking and data association*. Academic-Press, Boston, USA, 1988

[Bar-Shalom 2001] BAR-SHALOM, Y. ; LI, X. R. ; KIRUBARAJAN, T.: *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software.* Wiley & Sons, 2001

[Bar-Shalom 1975] BAR-SHALOM, Y. ; TSE, E.: Tracking in a Cluttered Environment with Probabilistic Data Association. In: *Automatica 11*, Nr. 5, 1975, S. 451–460

[Bartels 2009] BARTELS, A. ; STEINMEYER, S. ; SPICHALSKI, C.: Spurwechselassistenz. In: [Winner 2009c], S. 562–571

[Blackman 1999] BLACKMAN, S. ; POPOLI, R.: *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library, 1999

[Borgelt 2003] BORGELT, C. ; KLAWONN, F. ; KRUSE, R. ; NAUCK, D.: *Neuro-Fuzzy-Systeme*. Vieweg+Teubner, 2003

[Bouzouraa 2009] BOUZOURAA, M. E. ; HOFMANN, U.: Effiziente Kartenbasierte Umfeldwahrnehmung mit Laserscanner-Sensoren für Fahrerassistenzsysteme. In: STILLER, C. (Hrsg.) ; MAURER, M. (Hrsg.): *6. Workshop Fahrerassistenzsysteme*, Löwenstein/Hößlinsülz, 2009, S. 157–165

[Bouzouraa 2010] Bouzouraa, M. E. ; Hofmann, U.: Fusion of Occupancy Grid Mapping and Model based Object Tracking for Driver Assistance Systems using Laser and Radar Sensors. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, San Diego, USA, 2010, S. 294–300

[Brahmi 2013] Brahmi, M.: Reference Systems for Environmental Perception. In: Maurer, M. (Hrsg.) ; Winner, H. (Hrsg.): *Automotive Systems Engineering*. Springer, 2013, S. 205–221

[Bresenham 1965] Bresenham, J. E.: Algorithm for Computer Control of a Digital Plotter. In: *IBM Systems journal 4*, Nr. 1, 1965, S. 25–30

[Breuer 2009] Breuer, J.: Bewertungsverfahren von Fahrerassistenzsystemen. In: [Winner 2009c], S. 55–68

[Burgard 1998] Burgard, W. ; Cremers, A.B. ; Fox, D. ; Hähnel, D. ; Lakemeyer, G. ; Schulz, D. ; Steiner, W. ; Thrun, S.: The Interactive Museum Tour-guide Robot. In: *Proceedings of the National Conference on Artificial Intelligence*, Madison, USA, 1998, S. 11–18

[Burges 1998] Burges, C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data mining and knowledge discovery 2*, Nr. 2, 1998, S. 121–167

[Canny 1987] Canny, J.: A Computational Approach to Edge Detection. In: *Readings in computer vision: issues, problems, principles, and paradigms 184*, 1987, S. 87–116

[Caruana 2006] Caruana, R. ; Niculescu-Mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, USA, 2006, S. 161–168

[Chang 2011] Chang, C.-C. ; Lin, C.-J.: LIBSVM: a Library for Support Vector Machines. In: *ACM Transactions on Intelligent Systems and Technology 2*, 2011, S. 1–27

[Cooper 1992] Cooper, G. ; Herskovits, E.: A Bayesian Method for The Induction of Probabilistic Networks from Data. In: *Machine Learning 9*, 1992, S. 309–347

[Darms 2007] Darms, M.: *Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahrerassistenzsysteme*, Technische Universität Darmstadt, Diss., 2007

[Dasarathy 1990] DASARATHY, B. V.: *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1990

[Degerman 2012] DEGERMAN, P. ; AH-KING, J. ; NYSTRÖM, T. ; MEINECKE, M. ; STEINMEYER, S.: Targeting lane-change accidents for heavy vehicles. In: *VDI-Berichte* , Nr. 2166, 2012

[DeGroot 1982] DEGROOT, M. ; FIENBERG, S.: The Comparison and Evaluation of Forecasters. In: *The Statistician 32*, 1982, S. 12–22

[Dietmayer 2005a] DIETMAYER, K. ; KIRCHNER, A. ; KÄMPCHEN, N.: Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme. In: MAURER, M. (Hrsg.) ; STILLER, C. (Hrsg.): *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Springer, 2005, S. 661–692

[Dietmayer 2005b] DIETMAYER, K. ; KÄMPCHEN, N. ; FÜRSTENBERG, K. ; KIBBEL, J. ; JUSTUS, W. ; SCHULZ, R.: Roadway Detection and Lane Detection using Multilayer Laserscanner. In: VALLDORF, J. (Hrsg.) ; GESSNER, W. (Hrsg.): *Advanced Microsystems for Automotive Applications 2005*, Springer, 2005, S. 197–213

[Dornieden 2012] DORNIEDEN, B. ; JUNGE, L. ; PASCHEKA, P.: Vorausschauende Energieeffiziente Fahrzeuglängsregelung. In: *Automobiltechnische Zeitschrift 114*, Nr. 3, 2012

[Effertz 2009] EFFERTZ, J.: *Autonome Fahrzeugführung in urbaner Umgebung durch Kombination objekt– und kartenbasierter Umfeldmodelle*, Technische Universität Braunschweig, Diss., 2009

[Elfes 1989a] ELFES, A.: *Occupancy grids: A Probabilistic Framework for Robot Perception and Navigation*, Carnegie Mellon University Pittsburgh, USA, Diss., 1989

[Elfes 1989b] ELFES, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation. In: *IEEE Trans. on Computers 22*, Nr. 6, 1989, S. 46–57

[Elfes 1991] ELFES, A.: Dynamic Control of Robot Perception using Stochastic Spatial Models. In: *Proceedings of the International Workshop on Information Processing in Mobile Robots*, München, 1991, S. 203–218

[Elfes 1992] ELFES, A.: Dynamic Control of Robot Perception using Multi-property Inference Grids. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Nizza, Frankreich, 1992, S. 2561–2567

[Fan 2008] FAN, R.-E. ; CHANG, K.-W. ; HSIEH, C.-J. ; WANG, X.-R. ; LIN, C.-J.: LIBLINEAR: A Library for Large Linear Classification. In: *Journal of Machine Learning Research 9*, 2008, S. 1871–1874

[Faugeras 2004] FAUGERAS, O. ; LUONG, Q. ; PAPADOPOULLO, T.: *The Geometry of Multiple Images: the Laws that Govern the Formation of Multiple Images of a Scene and some of their Applications*. The MIT Press, 2004

[Fawcett 2004] FAWCETT, T.: ROC graphs: Notes and Practical Considerations for Researchers. In: *Machine Learning 31*, 2004, S. 1–38

[Ferguson 2006] FERGUSON, D. ; STENTZ, A.: Multi-resolution Field D*. In: *Proceedings of the International Conference on Intelligent Autonomous Systems*, Tokio, Japan, 2006, S. 65–74

[Gail 2010] GAIL, J.: Notbrems- und Spurverlassenswarnsysteme für schwere Nutzfahrzeuge - Technische Anforderungen aus Sicht der Gesetzgebung. In: [VDI Fahrzeug- und Verkehrstechnik 2010], S. 289–302

[Garey 1979] GAREY, M. R. ; JOHNSON, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979

[Gasser 2012] GASSER, T. ; ARZT, C. ; AYOUBI, M. ; BARTELS, A. ; EIER, J. ; FLEMISCH, F. ; HÄCKER, D. ; HESSE, T. ; HUBER, W. ; LOTZ, C. ; MAURER, M. ; RUTH-SCHUMACHER, S. ; SCHWARZ, J. ; VOGT, W.: Rechtsfolgen zunehmender Fahrzeugautomatisierung – Gemeinsamer Schlussbericht der Projektgruppe. In: *Berichte der Bundesanstalt für Straßenwesen*. Bergisch-Gladbach, 2012

[Geduld 2009] GEDULD, G.: Lidarsensorik. In: [Winner 2009c], S. 172–186

[Giachetti 1998] GIACHETTI, A. ; CAMPANI, M. ; TORRE, V.: The use of Optical Flow for Road Navigation. In: *IEEE Transactions on Robotics and Automation 14*, Nr. 1, 1998, S. 34–48

[Haines 1994] HAINES, E.: Point in Polygon Strategies. In: *Graphics Gems IV*, 1994, S. 24–46

[Hart 1968] Hart, P. E. ; Nilsson, N. J. ; Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: *IEEE transactions on Systems Science and Cybernetics 4*, Nr. 2, 1968, S. 100–107

[Haun 1998] Haun, M.: *Simulation Neuronaler Netze*. expert-Verlag, 1998

[Heckbert 1990] Heckbert, P. S.: A Seed Fill Algorithm. In: *Graphics Gems 1*, 1990, S. 275–277

[Hoeger 2011] Hoeger, R. ; Zeng, H. ; Hoess, A. ; Kranz, T. ; Boverie, S. ; Strauss, M. ; Jakobsson, E. ; Beutner, A. ; Bartels, A. ; To, T. ; Stratil, H. ; Fürstenberg, K. ; Ahlers, F. ; Frey, E. ; Schieben, A. ; Mosebach, H. ; Flemisch, F. ; Dufaux, A. ; Manetti, D. ; Amditis, A. ; Mantzouranis, I. ; Lepke, H. ; Szalay, Z. ; Szabo, B. ; Luithardt, P. ; Gutknecht, M. ; Schoemig, N. ; Kaussner, A. ; Nashashibi, F. ; Resende, P. ; Vanholme, B. ; Glaser, S. ; Allemann, P. ; Seglö, F. ; Nilsson, A.: Final Report / HAVEit Project (FP7/2007-2013 grant agreement n° 212154). 2011 (D61.1). – Public Deliverable

[ISO 2002] Norm ISO/IEC 15622:2002(E) 2002. *Transport Information and Control Systems – Adaptive Cruise Control Systems – Performance Requirements and Test Procedures*

[ISO 2008] Norm ISO/IEC 22179:2008(E) 2008. *Intelligent Transport Systems – Full Speed Range Adaptive Cruise Control (FSRA) Systems – Performance Requirements and Test Procedures*

[ISO 2011] Norm ISO/IEC 8855:2011(E) 2011. *Road vehicles – Vehicle dynamics and road-holding ability – Vocabulary*

[Jordan 2006] Jordan, R. ; Ahlrichs, U. ; Leineweber, T. ; Lucas, B. ; Knoll, P.: Hindernisklassifikation von stationären Objekten auf Basis eines nichtwinkeltrennfähigen Long-Range-Radar Sensors. In: Stiller, C. (Hrsg.) ; Maurer, M. (Hrsg.): *4. Workshop Fahrerassistenzsysteme*, Löwenstein/Hößlinsülz, 2006, S. 153–160

[Julier 1997] Julier, S. J. ; Uhlmann, J. K.: A new Extension of the Kalman Filter to Nonlinear Systems. In: *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing (AeroSense), Simulations and Controls*, Orlando, USA, 1997, S. 182–193

[Kalman 1960] Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transaction of ASME, Journal of Basic Engineering* , 1960, S. 25–45

[Kammel 2008] KAMMEL, S. ; ZIEGLER, J. ; PITZER, B. ; WERLING, M. ; GINDELE, T. ; JAGZENT, D. ; SCHRÖDER, J. ; THUY, M. ; GOEBL, M. ; HUNDELSHAUSEN, F. v. ; PINK, O. ; FRESE, C. ; STILLER, C.: Team AnnieWAY's autonomous system for the DARPA Urban Challenge 2007. In: *Journal of Field Robotics 25*, Nr. 9, 2008, S. 615–639

[Kelly 1998] KELLY, A. ; STENTZ, A.: Rough Terrain Autonomous Mobility – part 2: An active Vision, Predictive Control Approach. In: *Autonomous Robots 5*, Nr. 2, 1998, S. 163–198

[Kolski 2006] KOLSKI, S. ; FERGUSON, D. ; BELLINO, M. ; SIEGWART, R.: Autonomous Driving in Structured and Unstructured Environments. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, Tokio, Japan, 2006, S. 558–563

[Kompaß 2008] KOMPASS, K.: Fahrerassistenzsysteme der Zukunft – auf dem Weg zum autonomen Pkw? In: SCHINDLER, V. (Hrsg.) ; SIEVERS, I. (Hrsg.): *Forschung für das Auto von Morgen*, Springer, 2008, S. 261–285

[Konolige 1997] KONOLIGE, K.: Improved Occupancy Grids for Map Building. In: *Autonomous Robots 4*, Nr. 4, 1997, S. 351–367

[Kopetz 2011] KOPETZ, H.: *Real-time systems: design principles for distributed embedded applications*. Springer, 2011

[Kotsiantis 2007] KOTSIANTIS, S. B.: Supervised Machine Learning: A Review of Classification Techniques. In: *Informatika 31*, 2007, S. 249–268

[Lancaster 1963] LANCASTER, H.: Correlation and complete dependence of random variables. In: *The Annals of Mathematical Statistics 34*, Nr. 4, 1963, S. 1315–1321

[Ling 2002] LING, C. X. ; ZHANG, H.: The Representational Power of Discrete Bayesian Networks. In: *Journal of Machine Learning Research 3*, 2002, S. 709–721

[Lucas 1985] LUCAS, B. D.: *Generalized Image Matching by the Method of Differences*, Carnegie Mellon University, Diss., 1985

[Martin 1996] MARTIN, M. C. ; MORAVEC, H.P.: Robot Evidence Grids / Carnegie Mellon University, the Robotics Institute. 1996 (6). – Forschungsbericht

[Maurer 2004] MAURER, M. ; KIRCHNER, A. ; KOPISCHKE, S. ; MEINECKE, M. ; RANDLER, M. ; SCHOLZ, S. ; STÄMPFLE, M. ; WILHELM, U. ; ANKLAM, C. ; MEINKE, I. ; VON HOLT, V. ; KÜHNLE, G.: *Verfahren und Vorrichtung zur Erkennung von Hindernissen und/oder Verkehrsteilnehmern*. Deutsches Patent- und Markenamt, Bundesrepublik Deutschland, 2004. – DE A1 10 242 808

[Meinecke 2013] MEINECKE, M. ; STEINMEYER, S. ; DEGERMAN, P. ; AH-KING, J. ; NYSTRÖM, T. ; DEEG, C. ; MENDE, R.: Experiences with a radar-based side assist for heavy vehicles. In: *14th International Radar Symposium (IRS)* Bd. 2, Dresden, 2013, S. 720–725

[Meyer 1992] MEYER, F. ; BOUTHEMY, P.: Estimation of Time-to-collision Maps from First Order Motion Models and Normal Flows. In: *Proceedings of the 11th International Conference on Pattern Recognition and Information Processing* Bd. 1, Minsk, Weißrussland, 1992, S. 78–82

[Mählisch 2009] MÄHLISCH, M.: *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten*, Universität Ulm, Diss., 2009

[Mirwaldt 2012] MIRWALDT, P. ; BARTELS, A. ; TO, T. ; LEMMER, K. ; HUHLE, R. ; MALBERG, H. ; ZAUNSEDER, S.: Gestaltung eines Notfallassistenzsystems bei medizinisch bedingter Fahrunfähigkeit. In: *Beiträge zur 5. Tagung Fahrerassistenz*, München, 2012

[Montemerlo 2004] MONTEMERLO, M. ; THRUN, S.: A Multi-resolution Pyramid for Outdoor Robot Terrain Perception. In: *Proceedings of the 19th National Conference on Artifical intelligence*, San Jose, USA, 2004, S. 464–469

[Moravec 1985] MORAVEC, H. P. ; ELFES, A.: High Resolution Maps from Wide Angle Sonar. In: *Proceedings of the IEEE International Conference on Robotics and Automatisation*, St. Louis, USA, 1985, S. 116–121

[Muntzinger 2011] MUNTZINGER, M. M.: *Zustandsschätzung mit chronologisch ungeordneten Sensordaten für die Fahrzeugumfelderfassung*, Universität Ulm, Diss., 2011

[Munz 2011] MUNZ, M.: *Generisches Sensorfusionsframework zur gleichzeitigen Zustands- und Existenzschätzung für die Fahrzeugumfelderkennung*, Universität Ulm, Diss., 2011

[Mušicki 2004] MUŠICKI, D. ; EVANS, R.: Joint Integrated Probabilistic Data Association: JIPDA. In: *IEEE Transactions on Aerospace and Electronic Systems 40*, 2004

[Mušicki 1994] MUŠICKI, D. ; EVANS, R. ; STANKOVIC, S.: Integrated Probabilistic Data Association. In: *IEEE Transactions on Automatic Control 39*, 1994, S. 1237–1241

[Nissen 2003] NISSEN, S.: Implementation of a Fast Artificial Neural Network Library (FANN) / Department of Computer Science University of Copenhagen (DIKU). 2003. – Forschungsbericht. – http://fann.sf.net

[Noll 2009] NOLL, M. ; RAPPS, P.: Ultraschallsensorik. In: [Winner 2009c], S. 110–122

[Ohl 2011] OHL, S. ; MAURER, M.: Ein Kontur schätzendes Kalmanfilter mithilfe der Evidenztheorie. In: MAURER, M. (Hrsg.) ; DIETMAYER, K. (Hrsg.) ; FÄRBER, B. (Hrsg.) ; STILLER, C. (Hrsg.) ; WINNER, H. (Hrsg.): *7. Workshop Fahrerassistenzsysteme*, Walting/Altmühltal, 2011, S. 83–94

[Pearl 1988] PEARL, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988

[Petrovskaya 2009] PETROVSKAYA, A. ; THRUN, S.: Model based Vehicle Detection and Tracking for Autonomous Urban Driving. In: *Autonomous Robots 26*, 2009, S. 123–139

[Pinkus 1999] PINKUS, A.: Approximation Theory of the MLP Model in Neural Networks. In: *Acta Numerica 8*, 1999, S. 143–196

[Platt 1999] PLATT, J. C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: *Advances in large margin classifiers* , 1999, S. 61–74

[Polychronopoulos 2005a] POLYCHRONOPOULOS, A. ; AMDITIS, A. ; TSOGAS, M. ; ETEMAD, A.: Extended Path Prediction using Camera and Map Data for Lane Keeping Support. In: *Proceedings of the IEEE Intelligent Transportation Systems*, Wien, Österreich, 2005, S. 166–171

[Polychronopoulos 2005b] POLYCHRONOPOULOS, A. ; MÖHLER, N. ; GHOSH, S. ; BEUTNER, A.: System Design of a Situation Adaptive Lane Keeping Support System, the SAFELANE System. In: *Advanced Microsystems for Automotive Applications* , 2005, S. 169–183

[Rasshofer 2005] RASSHOFER, R. H. ; GRESSER, K.: Automotive Radar and Lidar Systems for Next Generation Driver Assistance Functions. In: *Advances in Radio Science 3*, 2005, S. 205–209

[Rojo 2007] ROJO, J. ; ROJAS, R. ; GUNNARSSON, K. ; SIMON, M. ; WIESEL, F. ; RUFF, F. ; WOLTER, L. ; ZILLY, F. ; SANTRAČ, N. ; GANJINEH, T. ; SARKOHI, A. ; ULBRICH, F. ; LATOTZKY, D. ; JANKOVIC, B. ; HOHL, G. ; WISSPEINTNER, T. ; MAY, S. ; PERVÖLZ, K. ; NOWAK, W. ; MAURELLI, F. ; DRÖSCHEL, D.: Spirit of Berlin: An Autonomous Car for the DARPA Urban Challenge - Hardware and Software Architecture / Freie Universität Berlin. 2007. – Forschungsbericht

[Rumelhart 1986] RUMELHART, D. E. ; HINTONT, G. E. ; WILLIAMS, R. J.: Learning Representations by Back–propagating Errors. In: *Nature 323*, 1986, S. 533–536

[Schabenberger 2007] SCHABENBERGER, R.: ADTF: Engineering-Framework for Driver Assistance and Safety Systems. In: *VDI Berichte 2000*, 2007

[Schiele 2009] SCHIELE, B. ; WOJEK, C.: Kamerabasierte Fußgängerdetektion. In: [Winner 2009c], S. 223–236

[Schoeberl 2008] SCHOEBERL, T. ; FOCKE, T. ; MÜLLER, M. ; HILSEBECHER, J. ; BECKER, J. ; SPARBERT, J. ; OECHSLE, F. ; ZENDER, A. ; SIMON, A. ; GRIMM, A.: *Fahrerassistenzsystem und Verfahren zur Objektplausibilisierung*. Deutsches Patent- und Markenamt, Bundesrepublik Deutschland, 2008. – DE A1 10 018 470

[Schuberth 2010] SCHUBERTH, S. ; DUBA, G.-P. ; SPARBERT, J. ; RAU, S.: Fahrschlauchfusion. In: [VDI Fahrzeug- und Verkehrstechnik 2010], S. 213–229

[Shafer 1976] SHAFER, G.: *A Mathematical Theory of Evidence*. Bd. 1. Princeton university press Princeton, 1976

[Sinha 2006] SINHA, A. ; KIRUBARAJAN, T. ; DING, Z. J. ; FAROOQ, M.: Track Quality based Multitarget Tracking Algorithm. In: *Proceedings of SPIE 6236*, 2006, S. 1–12

[Sittler 1964] SITTLER, R. W.: An Optimal Data Association Problem on Surveillance Theory. In: *IEEE Trans. on Military Electronics MIL-8*, 1964, S. 125–139

[Smarandache 2006] SMARANDACHE, Florentin ; DEZERT, Jean: *Advances and Applications of DSmT for Information Fusion*. 2nd. American Research Press, 2006

[Steinmeyer 2009] STEINMEYER, S.: Handling Contradictory Sensor Data in Environment Maps for Advanced Driver Assistance Systems. In: FISCHER, S. (Hrsg.) ; MAEHLE, E. (Hrsg.) ; REISCHUK, R. (Hrsg.): *Informatik 2009: Im Focus das Leben*, Köllen, Lübeck, 2009, S. 301

[Steinmeyer 2010a] STEINMEYER, S. ; BARTELS, A. ; MUSIAL, M. ; SCHULTZE, K. ; TO, T. ; WEISER, A.: Modulare Sensordatenfusions-Architektur am Beispiel des 'Temporary Auto Pilot'. In: *AAET 2010, Beiträge zum gleichnamigen 11. Braunschweiger Symposium*, Gesamtzentrum für Verkehr Braunschweig e. V., Braunschweig, 2010, S. 89–104

[Steinmeyer 2010b] STEINMEYER, S. ; MAURER, M.: Dynamische Bestimmung von Existenz- und Signifikanzwahrscheinlichkeiten für Umfeldobjekte. In: [VDI Fahrzeug- und Verkehrstechnik 2010], S. 197–212

[Steinwart 2008] STEINWART, I. ; CHRISTMANN, A.: *Support Vector Machines*. Springer, 2008

[Stüker 2004] STÜKER, D.: *Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr*, Carl von Ossietzky-Universität Oldenburg, Diss., 2004

[Sun 2006] SUN, Z. ; BEBIS, G. ; MILLER, R.: On-road Vehicle Detection: a Review. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, Nr. 5, 2006, S. 694–711

[Thomas 2007] THOMAS, P. ; FRAMPTON, R.: Real-world assessment of Relative Crash Involvement Rates of Cars equipped with Electronic Stability Control. In: *Proceedings of the 20th International Technical Conference on the Enhanced Safety of Vehicles Conference*, Lyon, Frankreich, 2007

[Thrun 1998] THRUN, S.: Learning Metric-topological Maps for Indoor Mobile Robot Navigation. In: *Artificial Intelligence 99*, Nr. 1, 1998, S. 21–71

[Thrun 2003] THRUN, S.: Learning Occupancy Grid Maps with Forward Sensor Models. In: *Autonomous robots 15*, Nr. 2, 2003, S. 111–127

[Thrun 2006] THRUN, S. ; MONTEMERLO, M. ; DAHLKAMP, H. ; STAVENS, D. ; ARON, A. ; DIEBEL, J. ; FONG, P. ; GALE, J. ; HALPENNY, M. ; HOFFMANN, G. ; LAU, K. ; OAKLEY, C. ; PALATUCCI, M. ; PRATT, V. ; STANG, P. ; STROHBAND, S. ; DUPONT, C. ; JENDROSSEK, L. ; KOELEN, C. ; MARKEY, C. ; RUMMEL, C. ; NIEKERK, J. v. ; JENSEN, E. ; ALESSANDRINI, P. ; BRADSKI, G. ; DAVIES, B. ; ETTINGER, S. ; KAEHLER, A. ; NEFIAN, A. ; MAHONEY, P.: Stanley, the Robot that won the DARPA Grand Challenge. In: *Journal of Field Robotics 23*, Nr. 9, 2006, S. 661–692

[To 2010a] TO, T. ; BARTELS, A.: Temporary Auto-Pilot: 1st System Functionality / HAVEit Project (FP7/2007-2013 grant agreement n$^\circ$ 212154). 2010 (D53.3). – Public Deliverable

[To 2010b] TO, T. ; KARRENBERG, S. ; STEINMEYER, S. ; WEISER, A. ; JOACHIM, M. ; JUNGE, L. ; BARTELS, A.: Temporary Auto-Pilot: Components installed, working and tested / HAVEit Project (FP7/2007-2013 grant agreement n$^\circ$ 212154). 2010 (D53.2). – Public Deliverable

[Urban 2006] URBAN, W. ; HENN, R. ; MOSER, V.: *Method for Determining the Plausibility of Objects in A Vehicle Driver Assistance System*. Europäisches Patentamt, 2006. – EP 1 853 453 B1

[Urmson 2008] URMSON, C. ; ANHALT, J. ; BAGNELL, D. ; BAKER, C. ; BITTNER, R. ; CLARK, M. N. ; DOLAN, J. ; DUGGINS, D. ; GALATALI, T. ; GEYER, C. ; GITTLEMAN, M. ; HARBAUGH, S. ; HEBERT, M. ; HOWARD, T. M. ; KOLSKI, S. ; KELLY, A. ; LIKHACHEV, M. ; MCNAUGHTON, M. ; MILLER, N. ; PETERSON, K. ; PILNICK, B. ; RAJKUMAR, R. ; RYBSKI, P. ; SALESKY, B. ; SEO, Y. ; SINGH, S. ; SNIDER, J. ; STENTZ, A. ; WHITTAKER, W. ; WOLKOWICKI, Z. ; ZIGLAR, J. ; BAE, H. ; BROWN, T. ; DEMITRISH, D. ; LITKOUHI, B. ; NICKOLAOU, J. ; SADEKAR, V. ; ; ZHANG, W. ; STRUBLE, J. ; TAYLOR, M. ; DARMS, M. ; FERGUSON, D.: Autonomous driving in urban environments: Boss and the urban challenge. In: *Journal of Field Robotics 25*, 2008, S. 425–466

[Uusitalo 2007] UUSITALO, L.: Advantages and challenges of Bayesian networks in environmental modelling. In: *Ecological modelling 203*, Nr. 3-4, 2007, S. 312–318

[van Dam 1996] VAN DAM, J. ; KRÖSE, B. ; GROEN, F.: Neural Network Applications in Sensor Fusion for An Autonomous Mobile Robot. In: *Reasoning with Uncertainty in Robotics* , 1996, S. 263–278

[VDI Fahrzeug- und Verkehrstechnik 2010] VDI FAHRZEUG- UND VERKEHRSTECHNIK (Hrsg.): *Beiträge zur 26. VDI/VW-Gemeinschaftstagung, Fahrerassistenz und Integrierte Sicherheit*. VDI Wissensforum GmbH, 2010

[von Zanten 2009] VON ZANTEN, A. ; KOST, F.: Bremsenbasierte Assistenzfunktionen. In: [Winner 2009c], S. 356–394

[Weiser 2009] WEISER, A. ; BARTELS, A. ; STEINMEYER, S. ; WEISS, K. ; MAREK, M.: Intelligent Car - Teilautomatisches Fahren auf der Autobahn. In: *AAET 2009, Beiträge zum gleichnamigen 10. Braunschweiger Symposium*, Gesamtzentrum für Verkehr Braunschweig e. V., Braunschweig, 2009, S. 11–26

[Weiser 2011] WEISER, A. ; TO, T.: *Verfahren zur Zuordnung von Fahrstreifen zu einem Fahrzeug*. Deutsches Patent- und Markenamt, Bundesrepublik Deutschland, 2011. – DE 10 2009 060 600.9

[Weiss 2008] WEISS, T. ; DIETMAYER, K.: Positionierung eines Fahrzeugs in unbekannten Gebieten mit Hilfe von Laserscannern (Localization in Unknown Environments Using Laser Scanner). In: *at–Automatisierungstechnik 56*, Nr. 11, 2008, S. 554–562

[Winner 2009a] WINNER, H.: Radarsensorik. In: [Winner 2009c], S. 123–171

[Winner 2009b] WINNER, H. ; DANNER, B. ; STEINLE, J.: Adaptive Cruise Control. In: [Winner 2009c], S. 478–521

[Winner 2009c] WINNER, H. (Hrsg.) ; HAKULI, S. (Hrsg.) ; WOLF, G. (Hrsg.): *Handbuch Fahrerassistenzsysteme*. Vieweg+Teubner, 2009

[Yu 2007] YU, H. ; WANG, Y. ; PENG, J.: An Occupancy Grids Building Method with Sonar Sensors Based on Improved Neural Network Model. In: *Proceedings of the 4th International Symposium on Neural Networks*, Nanjing, China, 2007, S. 592–601

[Zabler 2010] ZABLER, E. ; FINGBEINER, S. ; WELSCH, W. ; KITTEL, H. ; BAUER, C. ; NOETZEL, G. ; EMMERICH, H. ; HOPF, G. ; KONZELMANN, U. ; WAHL, T. ; NEUL, R. ; MÜLLER, W. ; BISCHOFF, C. ; PFAHLER, C. ; WEIBERLE, P. ; PAPERT, U. ; GERHARDT, C. ; MIEKLEY, K. ; FREHOFF, R. ; MAST, M. ; BAUER, B. ; HARDER, M. ; KASTEN, K. ; BRENNER, P. ; WOLF, F. ; RIEGEL, J. ; ARNDT, M. ; REIF, K. (Hrsg.): *Sensoren im Kraftfahrzeug*. Vieweg+Teubner, 2010